LUCA NOVELLI

IL MIO PRIMO LIBRO DI BASIC

E' IL PIU' DIFFUSO LINGUAGGIO DEI COMPUTER!

QUESTO LIBRO E' UN PO' COME UN VOCABOLARIO.

PER OGNI "PAROLA" C'E' UN ESERCIZIO DIVERTENTE!

E' FACILE IMPARARE!

VI SERVIRA' ANCHE A SCUOLA

E' INTERNAZIONALE!

E POI CI SONO ANCH'IO...

!

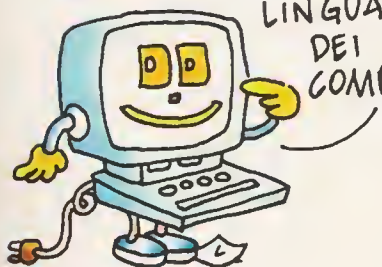
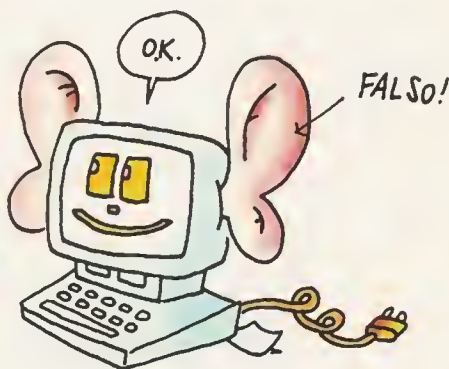
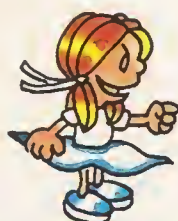
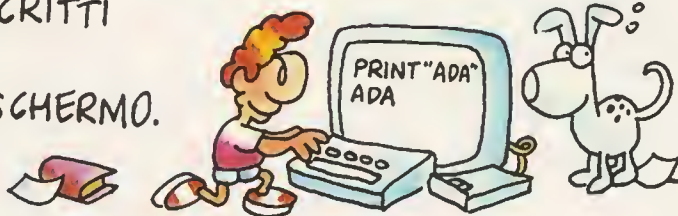
Edizioni Elettroniche Serie LIBRI

LUCA NOVELLI

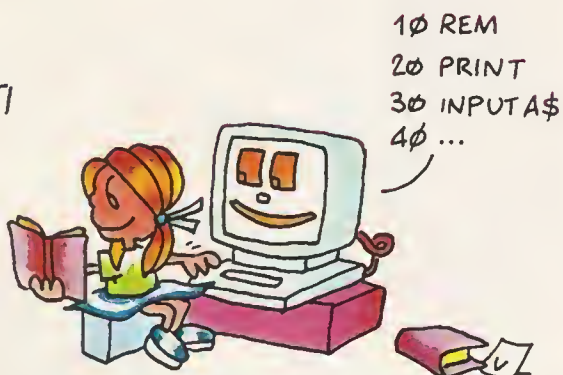
IL MIO PRIMO LIBRO DI

ARNOLDO MONDADORI EDITORE

Premessa

Dove si scopre
che il BASIC
non è una lingua
extraterrestre

La sigla BASIC sta per Beginners' All-purpose Symbolic Instruction Code (Codice di istruzioni di uso universale per principianti). Comprende un vocabolario di circa 200 termini, la maggior parte dei quali sono molto comuni nella lingua inglese come AND (= e), PRINT (= stampa), GOTO (= vai a), eccetera, eccetera.

PROGRAMMI BREVISSIMI
POSSONO SVOLGERE "LAVORI"
MOLTO LUNGHI E COMPLICATI.
ANCHE UN VIDEOGIOCO
E IL PIANO DI VOLO
DELLO SPACE SHUTTLE
SONO "PROGRAMMI".

CON QUESTO LIBRO
IMPARERETE I PIÚ IMPORTANTI
VOCABOLI BASIC
E POTRETE FAR "SVOLGERE"
AL VOSTRO COMPUTER
ALTRETTANTI PROGRAMMI
BASIC.

10 REM
20 PRINT
30 INPUT A$
40 ...

QUANDO AVRETE TRASCRITTO UN PROGRAMMA POTRETE CONSERVARLO E UTILIZZARLO TUTTE LE VOLTE CHE VORRETE.

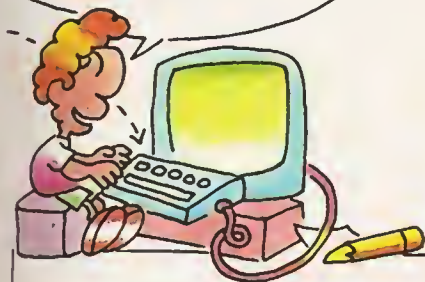
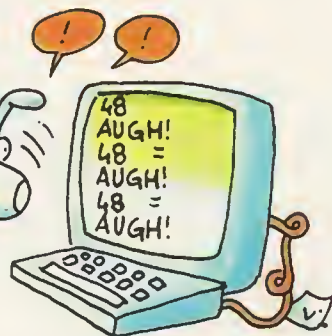
GUAI A SBAGLIARE, PERO! IL COMPUTER E' PEDANTE E NOIOSO. SE SBAGLIATE UN VOCABOLO O DIMENTICATE UNA VIRGOLETTA IL COMPUTER SI RIFIUTERA' DI ESEGUIRE I VOSTRI ORDINI. E NELLA MIGLIORE DELLE IPOTESI VI FARA' SUBITO NOTARE L'ERRORE "DI SINTASSI".

UN CONSIGLIO: E' INUTILE ARRABBIARSI CON UN COMPUTER

E' UNA MACCHINA CHE FARA' SEMPRE E SOLO QUELLO CHE GLI DITE DI FARE, IN BASIC, NATURALMENTE.

LA GRAMMATICA DEL BASIC

Dove scopriamo che il computer obbedisce anche al punto e virgola

COL PROGRAMMA, INSIEME ALLE ISTRUZIONI, DATE AL COMPUTER ANCHE PAROLE E NUMERI.

STRINGA. E' una sequenza di caratteri (parole, lettere dell'alfabeto, simboli grafici e anche spazi vuoti) posta tra virgolette. "AUGH!" è una stringa di cinque caratteri.
" " LE VIRGOLETTE (o apici). Servono al computer per riconoscere l'inizio e la fine di una stringa da stampare o da trattare in un altro modo.
COSTANTI. Sono i caratteri tra le virgolette e i valori numerici che nel programma rimangono invariati.

IL COMPUTER RICONOSCE ANCHE ALTRI SEGNI DI PUNTEGGIATURA BASIC.

; IL PUNTO E VIRGOLA. Ordina al computer di stampare di seguito anche il contenuto della successiva istruzione PRINT.
, VIRGOLA. Ordina al computer di incolonnare i dati da stampare. Serve anche per dividere le variabili in entrata.
: DUE PUNTI. Se alla fine di una istruzione ci sono i due punti si può aggiungere un'altra istruzione sulla stessa riga.
. IL PUNTO. Divide le unità dai decimali. Per esempio il numero 167,61 per il computer deve essere scritto 167.61.

MA NON DIMENTICATE CHE IL COMPUTER E' SOPRATUTTO UN CALCOLATORE.

PERCIO' OBBEDIRA' A TUTTI GLI ORDINI DI CALCOLO CHE RICEVE.

ANCHE QUESTI PER IL COMPUTER SONO ORDINI IN BASIC.

SEGNI DI ADDIZIONE E SOTTRAZIONE

SEGNO DI ELEVAZIONE A POTENZA. NOI USEREMO IL SEGNO ^. IN ALTRE TASTIERE SI USANO DUE SEGNI **.

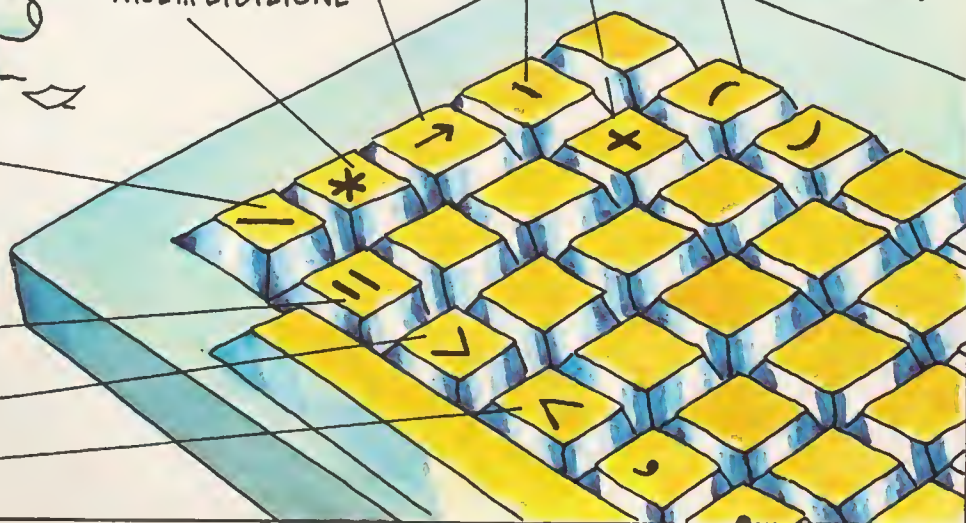
PARENTESI

SEGNO DI MOLTIPLICAZIONE

SEGNO DI DIVISIONE

SEGNO DI UGUALE

MAGGIORE

MINORE

MEMORIE. Il "cervello" del computer ha una parte logica (CPU) che sa contare e mettere in ordine i dati, e due memorie, la ROM e la RAM. La ROM contiene le istruzioni che fanno funzionare il computer, per esempio sa "riconoscere" il BASIC. Le celle della RAM invece ospitano i dati e le istruzioni dei vostri programmi.

PER IL COMPUTER UN PROGRAMMA E' UN PO' COME IL GIOCO DELL'OCA.
RUN
QUANDO GLI DATE L'ORDINE DI PARTIRE COMINCIA A ESEGUIRE TUTTE LE ISTRUZIONI CHE INCONTRA, RIGA DOPO RIGA.
10 REM
GIOCO DEL PROGRAMMA
20 PRINT
"QUANTI ANNI HAI?"
ALLA RIGA 10 INCONTRA REM (=ANNOTA) DA' UN'OCCHIATA E PASSA OLTRE.
ALLA RIGA 20 INCONTRA PRINT (= STAMPA) E SCRIVE SULLO SCHERMO TUTTO QUELLO CHE C'E' TRA LE VIRGOLETTE.
70 GOTO 90
ALLA RIGA 60 TROVA L'ORDINE DI SCRIVERE LA FRASE TRA LE VIRGOLETTE E POI AVANZA ALLA 70.
QUI TROVA L'ORDINE GOTO (=VAI A) 90 E SUBITO SALTA ALLA RIGA 90.
80 PRINT
"MI PIACCIONO I GIOVANISSIMI"
MI PIACCIONO I GIOVANISSIMI!
QUI SCRIVE LA FRASE TRA LE VIRGOLETTE E POI PROSEGUE ALLA 90
90 END
QUANDO ARRIVA A END (= FINE) SI FERMA: IL PROGRAMMA E' FINITO.

QUANTI ANNI HAI ?
?
ALLA RIGA 4Ø IL COMPUTER TROVA L'ORDINE LET (PONI) B = 18 E ASSEGNA IL VALORE 18 ALLA "VARIABILE" B.
NPUT A
A
40 LET B=18
18
B
QUI TROVA INPUT (=METTI DENTRO) E ASPETTA DA VOI UN NUMERO CHE ASSEGNERA' ALLA "VARIABILE" A.
50 IF A<B THEN 80
18
B
A
60 PRINT "SIAMO VECCHIETTI, EH?"
ALLA RIGA 5Ø IL COMPUTER DEVE FARE UNA SCELTA
SE (IF) IL CONTENUTO DI A E' MINORE (<) DI B ALLORA (THEN) DEVE ANDARE ALLA RIGA 8Ø.
ALTRIMENTI PROSEGUE ALLA RIGA SUCCESSIVA, LA 6Ø
©LUCA NOVELLI

DIAGRAMMA DI FLUSSO. Lo chiamano anche flow chart o diagramma a blocchi. È la forma grafica piu semplice per descrivere lo schema di un programma. Ogni blocco darà luogo a una riga di istruzioni. Quelli dove il computer deve prendere una decisione sono rappresentate da un rombo. Le istruzioni GO TO (vai a) sono rappresentate solo da una freccia. In questo libro sono riportati anche i flow chart di alcuni programmi, così potrete scoprire come sono stati costruiti.

10 START

20 PRINT"QUANTI ANNI HAI?"

30 INPUT A

40 LET B =18

50 IF A<B THEN

SE E' VERO

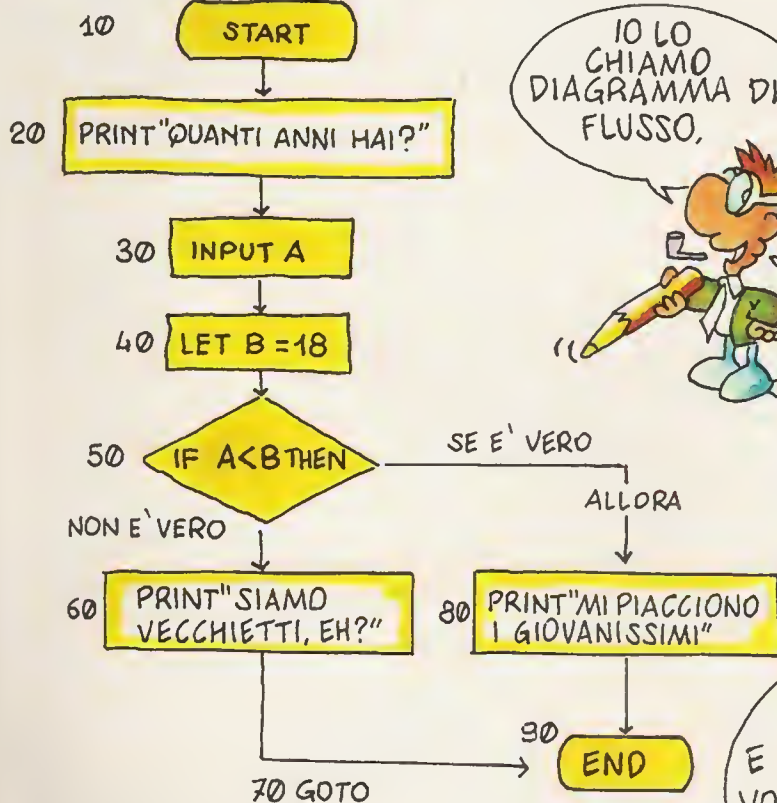
ALLORA

NON E' VERO

60 PRINT"SIAMO VECCHIETTI, EH?"

80 PRINT"MI PIACCIONO I GIOVANISSIMI"

90 END

70 GOTO

IO LO CHIAMO DIAGRAMMA DI FLUSSO.

FARE UN DIAGRAMMA DI FLUSSO VUOL DIRE DIVIDERE UN PROBLEMA IN TANTI PICCOLI PASSI CHE ANCHE UN TESTONE DI COMPUTER PUO' CAPIRE!

MA ADESSO BASTA CON I PREAMBOLI E SOTTO CON IL NOSTRO VOCABOLARIETTO DI BASIC! COMINCIAMO DA AND!

EH!

EH!

20 PRUNT

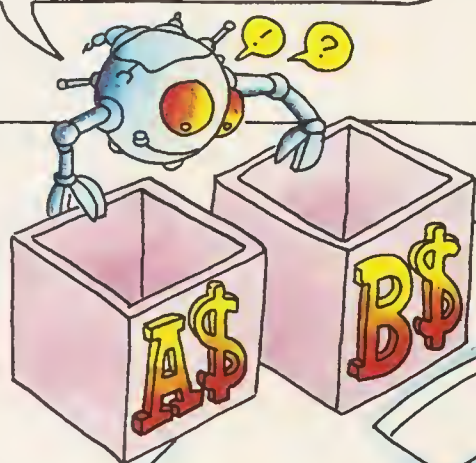
E ATTENTI AGLI ERRORI...

...CHE GLI ESPERTI CHIAMANO BUG(=INSETTI)

AND (=e). È una "congiunzione" anche in BASIC, il computer obbedisce all'istruzione solo se le condizioni poste per A AND (e) B sono vere. In tutti gli altri casi non fa nulla e procede alla riga successiva.

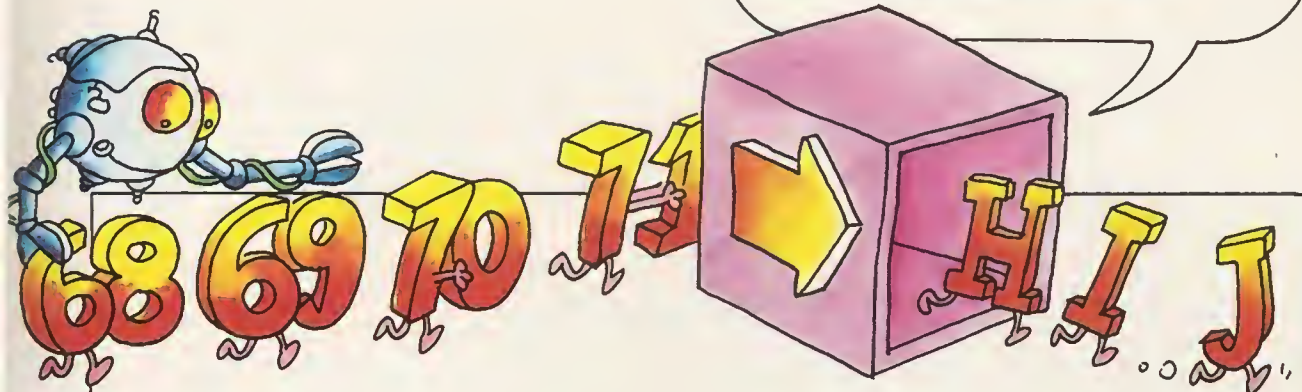
PROGRAMMINO DI BENVENUTO

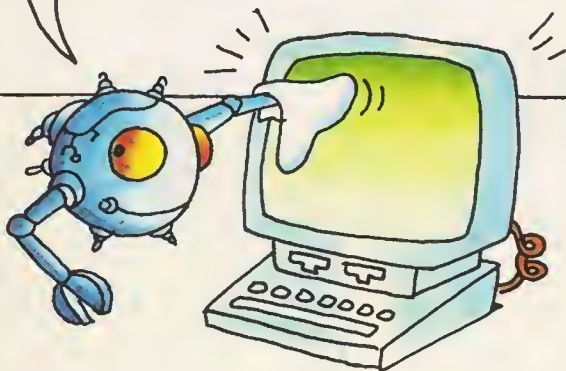
Molti ragazzi vengono a trovare Marko per vedere e provare il suo computer. Allora Marko ha scritto questo piccolo programma per dare Il benvenuto ai suoi amici. Ma solo se A$="SI" e B$="SI" il computer farà gli onori di casa, altrimenti...

```
 10 REM PROGRAMMA DI BENVENUTO
 20 PRINT"CIAO SONO IL COMPUTER DI MARCO"
 30 PRINT"TU CHI SEI ?"
 40 INPUT N$
 50 PRINT"CIAO ";N$;" PIACERE DI CONOSCERTI"
 60 PRINT"TI PIACCIONO I COMPUTER ?"
 70 PRINT"RISPONDI SI O NO"
 80 INPUT A$
 90 IF A$="SI" OR A$="NO" GOTO 110
100 GOTO 70
110 PRINT"VUOI IMPARARE IL BASIC ?"
120 PRINT"RISPONDI SI O NO"
130 INPUT B$
140 IF B$="SI" OR B$="NO" GOTO 160
150 GOTO 120
160 IF A$="SI" AND B$="SI" GOTO 180
170 PRINT"VAI A FARTI FRIGGERE ! ";N$:END
180 PRINT"BENVENUTO TRA NOI ";N$
```

ASC (da ASCII). Dice al computer di convertire un carattere (una lettera dell'alfabeto, un simbolo, eccetera) in un numero. Non sarà un numero qualunque ma il numero corrispondente del codice ASCII. Non è un vocabolo BASIC universale, lo Spectrum invece di ASC usa CODE.

ALLA SCOPERTA DEL CODICE ASCII

ASCII deriva dalle iniziali di American Standard Code for Information Interchange (codice americano standard per lo scambio delle informazioni). È un codice ormai internazionale che rappresenta lettere e numeri con sequenze di otto bit (zero e uno). Le sequenze possibili di otto bit sono 256 (2^8) e possono perciò codificare 256 caratteri, simboli grafici eccetera. I numeri che scoprirete con il nostro esercizio sono la forma decimale di queste sequenze di numeri binari. Per esemplo il numero di codice ASCII della A è 65, e rappresenta in forma decimale il numero binario 01000001.

CHR$ (da character=carattere). Converte numeri in caratteri. Compie l'operazione inversa di ASC. Per esempio PRINT CHR$ (65) visualizzerà la lettera A, come previsto dalla tabella del codice ASCII.

```
 10 REM IL CODICE ASCII
 20 CLS:PRINT:PRINT
 30 PRINT"PREMI :":PRINT
 40 PRINT"    1    PER VEDERE LA TABELLA ASCII":PRINT
 50 PRINT"    2    PER VEDERE LA CORRISPONDENZA"
 60 PRINT"         CARATTERE - CODICE":PRINT
 70 PRINT"    3    PER VEDERE LA CORRISPONDENZA"
 80 PRINT"         CODICE - CARATTERE":PRINT
 90 PRINT"    4    PER FINIRE"
100 INPUT S
110 ON S GOTO 130,190,290,410
120 CLS
130 FOR I=1 TO 255
140 PRINT CHR$(I),I
150 NEXT I
160 PRINT"PREMI RETURN PER CONTINUARE"
170 INPUT A$
180 IF A$=""GOTO 20
190 CLS:PRINT
200 PRINT"IMMETTI IL CARATTERE DI CUI VUOI VEDERE IL CODICE"
210 INPUT C$
220 PRINT:PRINT
230 PRINT"AL CARATTERE ";C$;" CORRISPONDE IL CODICE ASCII";ASC(C$)
240 PRINT:PRINT
250 PRINT"VUOI VEDERNI UN ALTRO ?"
260 INPUT A$
270 IF A$="SI" GOTO 190
280 GOTO 20
290 CLS:PRINT
300 PRINT"IMMETTI IL CODICE (COMPRESO TRA 1 E 255) DI CUI"
310 PRINT"VUOI VEDERE IL CARATTERE CORRISONDENTE"
320 INPUT C
330 IF C<1 OR C>255 GOTO 290
340 PRINT:PRINT
350 PRINT"AL CODICE ";C;"CORRISPONDE IL CARATTERE ";CHR$(C)
360 PRINT:PRINT
370 PRINT"VUOI VEDERNE UN ALTRO ?"
380 INPUT A$
390 IF A$="SI" GOTO 290
400 GOTO 20
410 CLS:END
```

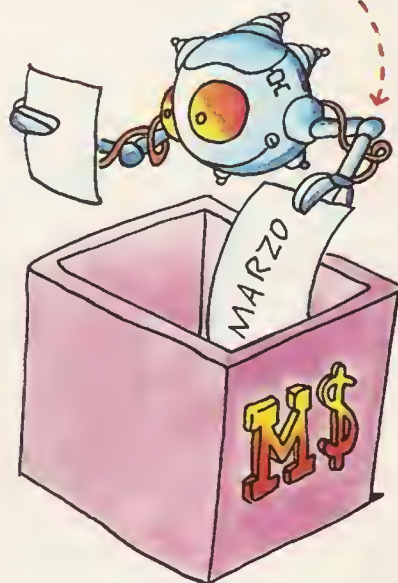
CLS (da CLear the Screen=sgombra lo schermo). Dice al computer di togliere tutto quello che c'è sullo schermo. Non è un vocabolo BASIC universale, per esempio Apple ha HOME, VIC 20 e COMMODORE 64 hanno PRINT "{SHIFT CLR HOME}".

IL SEGRETO DEI VIDEOGIOCHI

Il comando CLS (o gli equivalenti) sono il segreto dell'animazione nei videogiochi. Il principio è molto semplice, si da al computer l'ordine di stampare (PRINT) un punto o un simbolo grafico, subito dopo gli si da l'ordine di cancellarlo (CLS) e poi di ristamparlo un po' più in là. Così ai nostri occhi si crea l'illusione ottica che il punto si muova sullo schermo. In realtà sono tanti nuovi punti che subito dopo vengono cancellati.

DATA (=dati). Prepara dati, numeri o stringhe. Questi dati saranno poi associati alle variabili indicate da una successiva istruzione READ (=leggi). Una terza istruzione, RESTORE (=ripristina) dirà al computer di rimettere tutto nelle condizioni iniziali di DATA.

I MESI DELL'ANNO

DATA si trova sempre insieme a READ e RESTORE formando un'unica istruzione di assegnazione. Nel nostro esempio DATA "prepara" i mesi dell'anno che poi sono associati da READ alla variabile M$. La terza istruzione, RESTORE, dirà al computer di ripristinare tutto com'era alla prima riga di DATA, permettendovi di continuare.

```
 10 DATA GENNAIO,FEBBRAIO,MARZO,APRILE
 20 DATA MAGGIO,GIUGNO,LUGLIO,AGOSTO
 30 DATA SETTEMBRE,OTTOBRE,NOVEMBRE,DICEMBRE
 40 CLS:PRINT"DAMMI UN MESE IN NUMERO"
 50 INPUT M
 60 M=INT(M)
 70 IF M<1 OR M>12 GOTO 40
 80 FOR X=1 TO M
 90 READ M$
100 NEXT X
110 PRINT:PRINT M$
120 RESTORE
130 PRINT:PRINT"VUOI CONTINUARE ?"
140 INPUT A$
150 IF A$="SI" GOTO 40
160 END
```

DEF FN. Permette di definire nuove funzioni matematiche e di racchiuderle appunto nella funzione FN. In caso di ripetizioni, per esempio, permette di risparmiare tempo e diminuire la possibilità di errori.

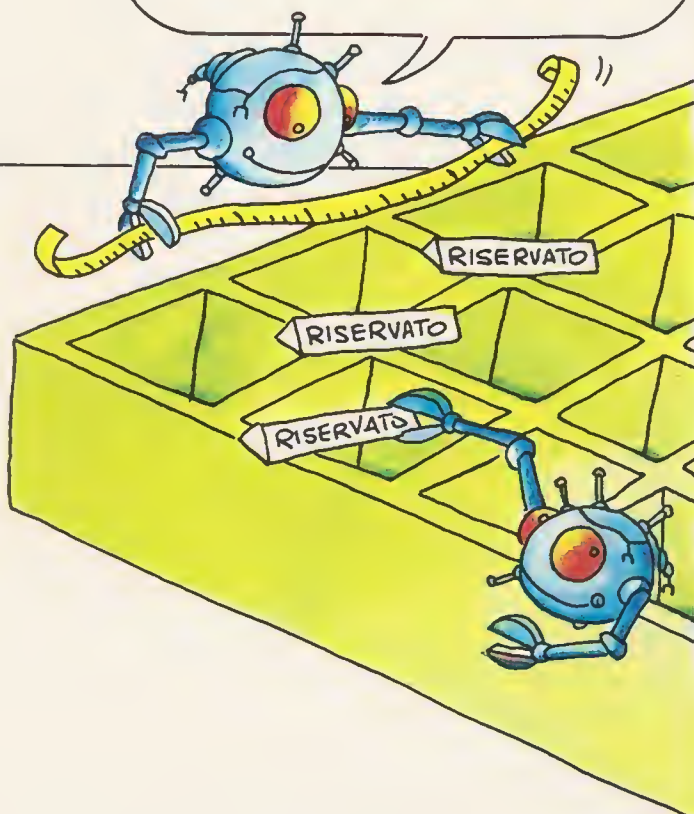
```
 10 REM DEFINISCO UNA FUNZIONE COMPLICATA
 20 DEF FNY(X)=(SIN(X)*X^2*(COS(2*X)+X^3)-X^5)/(X^2+1)
 30 REM ORA LA FUNZIONE PUO' ESSERE USATA QUANTO
 40 REM SI VUOLE ANCHE PIU' VOLTE NELLO STESSO PROGRAMMA
 50 CLS
 60 PRINT:PRINT"IMMETTI UN VALORE PER LA VARIABILE X"
 70 INPUT X
 80 Y=FNY(X)
 90 Z=(FNY(X))^2
100 PRINT"X = ";X,"Y = ";Y,"Z = ";Z
110 PRINT:PRINT"VUOI CALCOLARE ALTRI VALORI ?"
120 INPUT A$
130 IF A$="SI" GOTO 60
140 END
```

DIM (da DIMension=dimensione). Dice al computer di dare le dimensioni ad un opportuno spazio di memoria e di riservarlo a stringhe di caratteri, numeri o intere tabelle (matrici) che non potrebbero essere contenute in una singola "variabile".

DIM

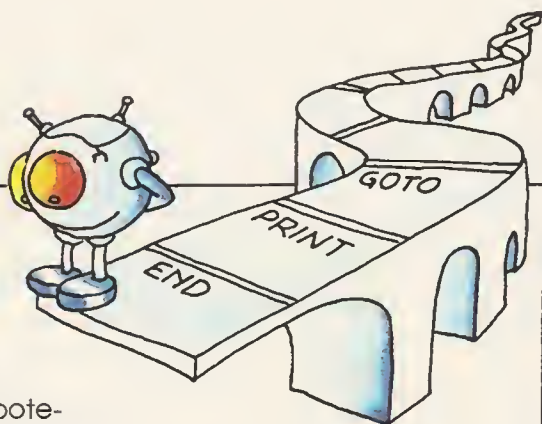
RISERVATO

RISERVATO

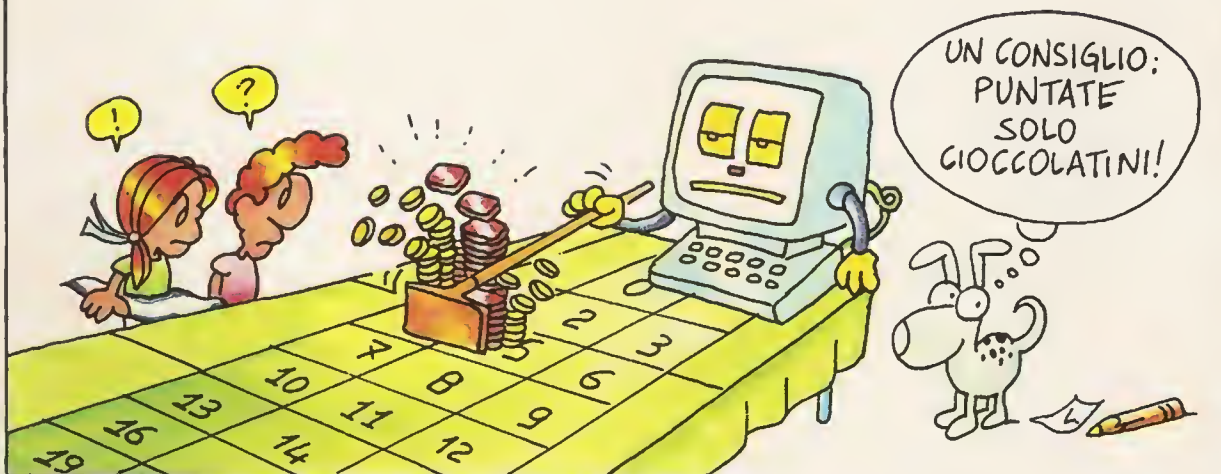
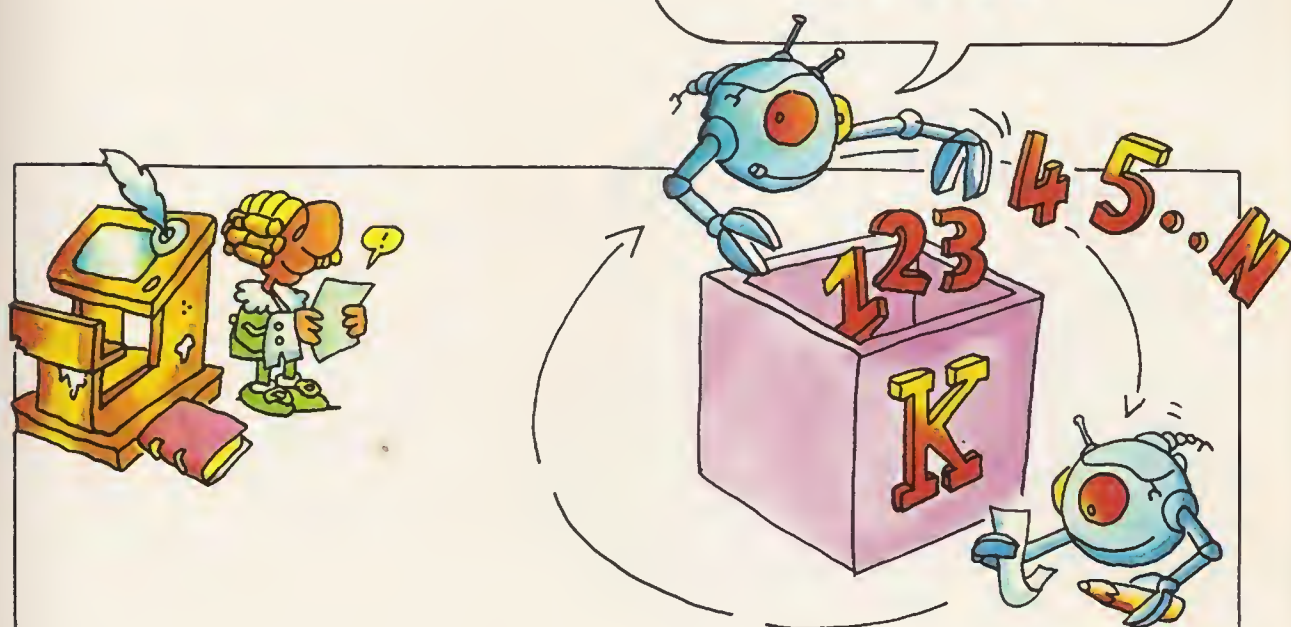
RISERVATO

VOLETE FAR SCRIVERE LE TABELLINE AL VOSTRO COMPUTER?

Con questo programmino il computer scriverà sullo schermo la tabellina che desiderate, fino a quella del 10.

```
 10 REM TABELLINE
 20 DIM T(10,10)
 30 FOR J=1 TO 10
 40 FOR K=1 TO 10
 50 T(J,K)=J*K
 60 NEXT K
 70 NEXT J
 80 CLS:PRINT"CHE TABELLINA VUOI VEDERE ?"
 90 INPUT I
100 IF I<0 OR I>10 GOTO 80
110 PRINT
120 FOR L=1 TO 10
130 PRINT T(I,L)
140 NEXT L
150 PRINT:PRINT"VUOI VEDERNE UN' ALTRA ?"
160 INPUT A$
170 IF A$<>"SI" AND A$<>"NO" GOTO 150
180 IF A$="SI" GOTO 80
190 END
```

END (=fine). Dice al computer che il programma è finito e di non fare più nulla. Nel linguaggio BASIC è indispensabile solo in qualche caso. Alcuni computer invece di END usano STOP.

LA ROULETTE ELETTRONICA

Siete fortunati al casinò? Con questo programmino potete verificarlo. Infatti il computer estrarrà un numero tra O e 36. Buona fortuna.

```
10 REM ROULETTE ELETTRONICA
20 N=INT(RND*(37))
30 CLS
40 PRINT:PRINT"PUNTA SU UN NUMERO TRA 1 E 36"
50 INPUT T
60 IF T=N THEN PRINT"HAI VINTO !":END
70 PRINT"HAI PERSO"
80 GOTO 40
90 END
```

FOR/TO/NEXT (=per/al/prossimo). Il computer eseguirà le istruzioni poste tra FOR, TO e NEXT finchè la "variabile di ciclo" non avrà raggiunto il valore indicato. Nel nostro esempio la "variabile di ciclo" è K e dovrà raggiungere il valore N che voi inserirete.

UN PROBLEMINO PER GAUSS

Il giovanissimo Karl Friedrich Gauss era destinato a diventare un grande matematico. Da piccolo si trovò davanti un maestro molto antipatico che per castigo gli assegnò il compito di sommare tutti i numeri interi compresi tra 1 e 100. Gauss lo risolse con una sola operazione. Voi potreste farlo risolvere al computer con questo programmino.

```
10 CLS:PRINT"SOMMO I NUMERI FINO A ?"
20 INPUT N
30 S=0
40 FOR K=1 TO N
50 S=S+K
60 NEXT K
70 PRINT:PRINT"LA SOMMA DEI PRIMI ";N;" INTERI VALE :";S
80 END
```

FOR TO NEXT

ECCO COME E' STATO COSTRUITO IL PROGRAMMA PER IL VOSTRO COMPUTER.

START

INPUT N

S=0
K+1

K>N

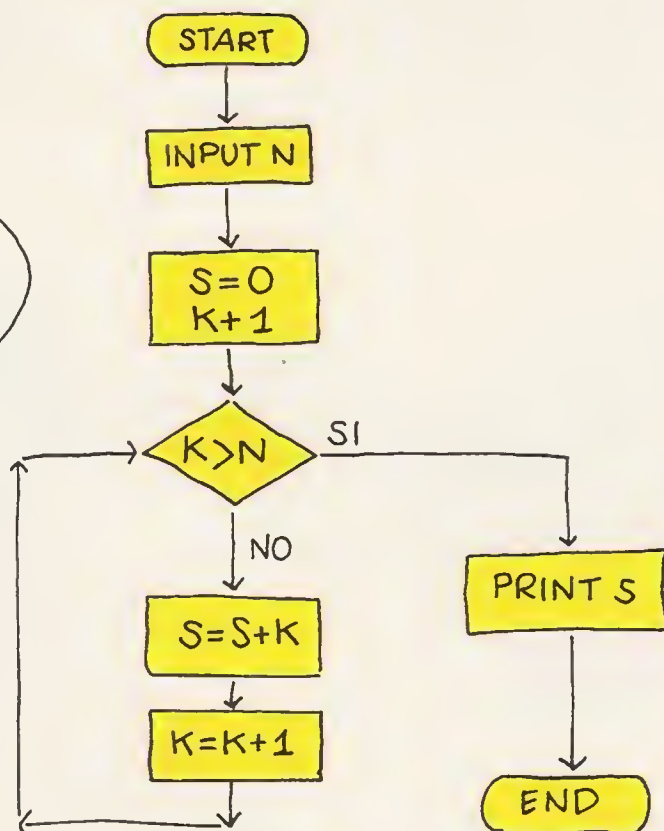
SI

PRINT S

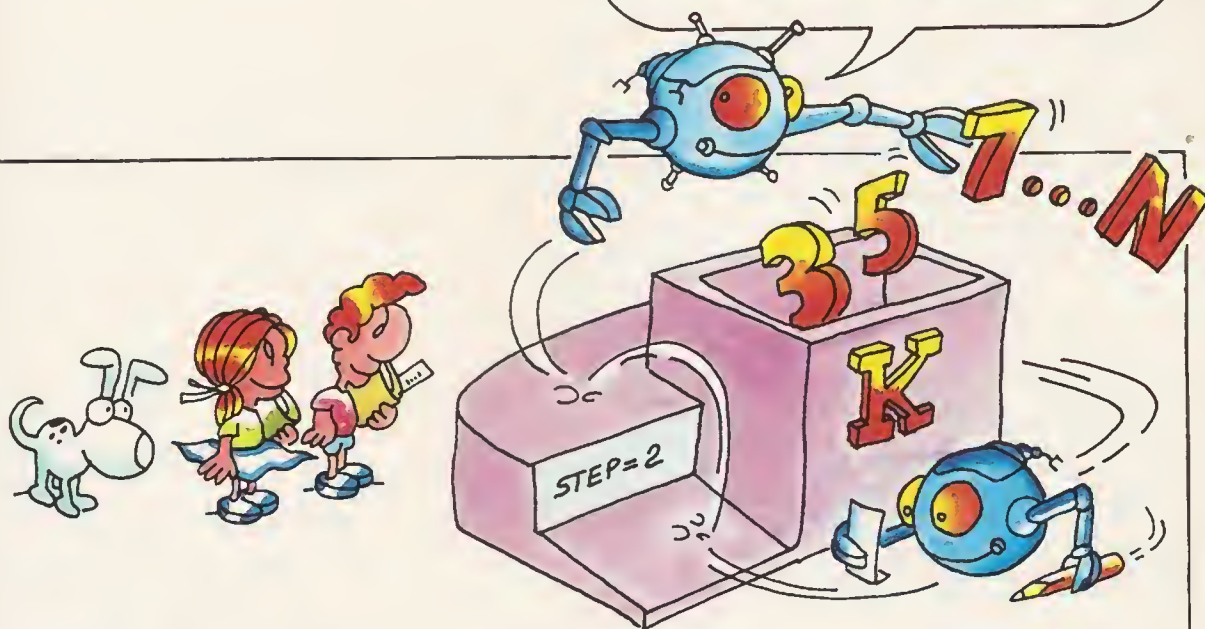
END

NO

S=S+K

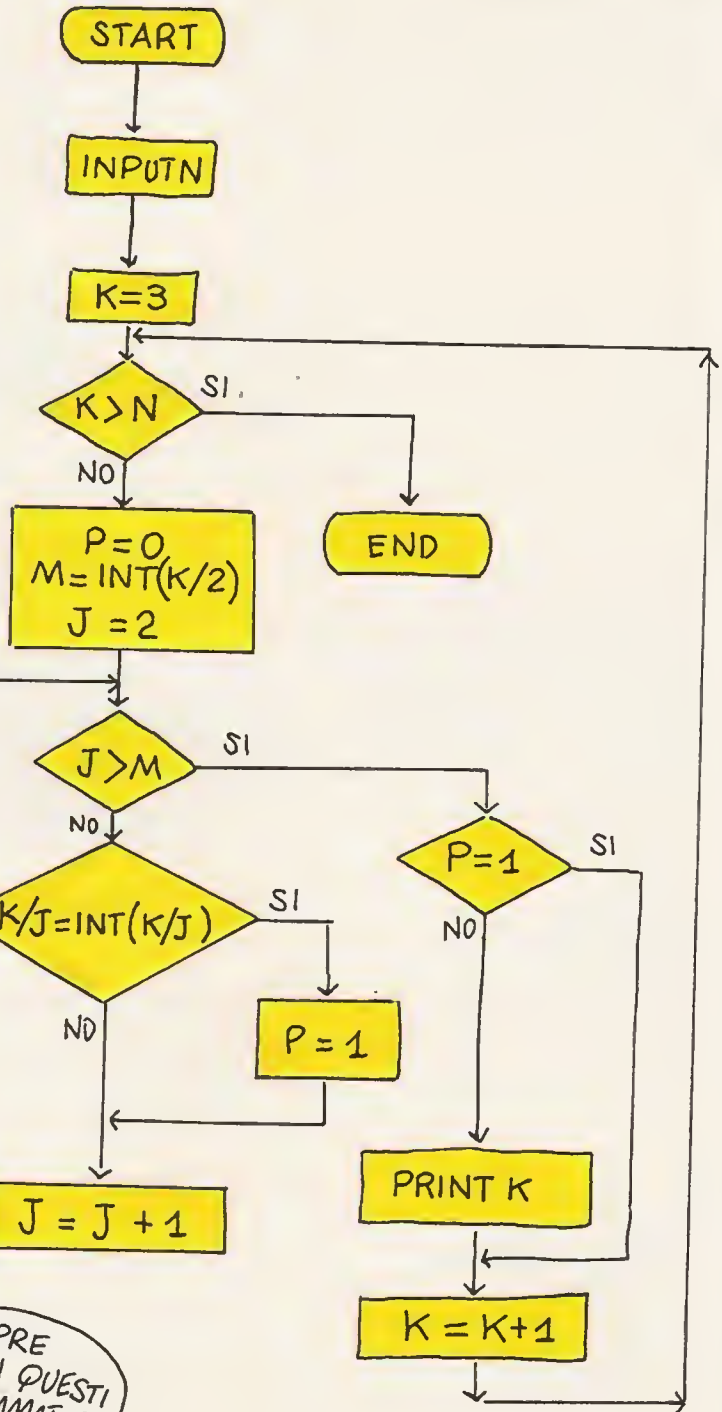
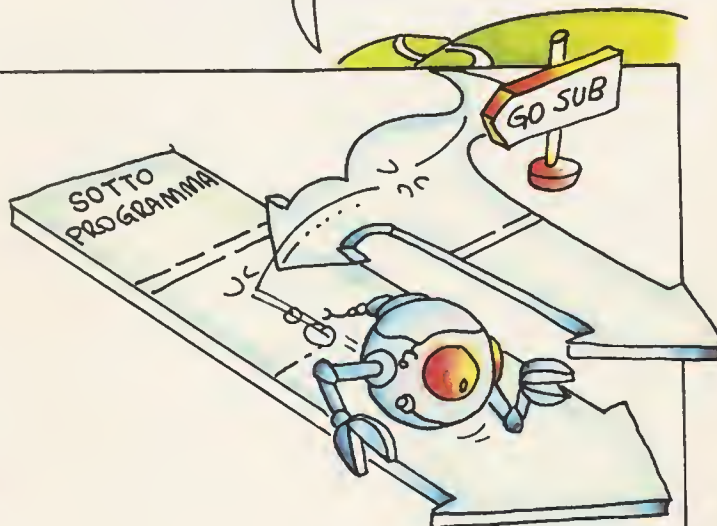
K=K+1

E IL PICCOLO GAUSS
COME RISOLSE IL SUO PROBLEMA?

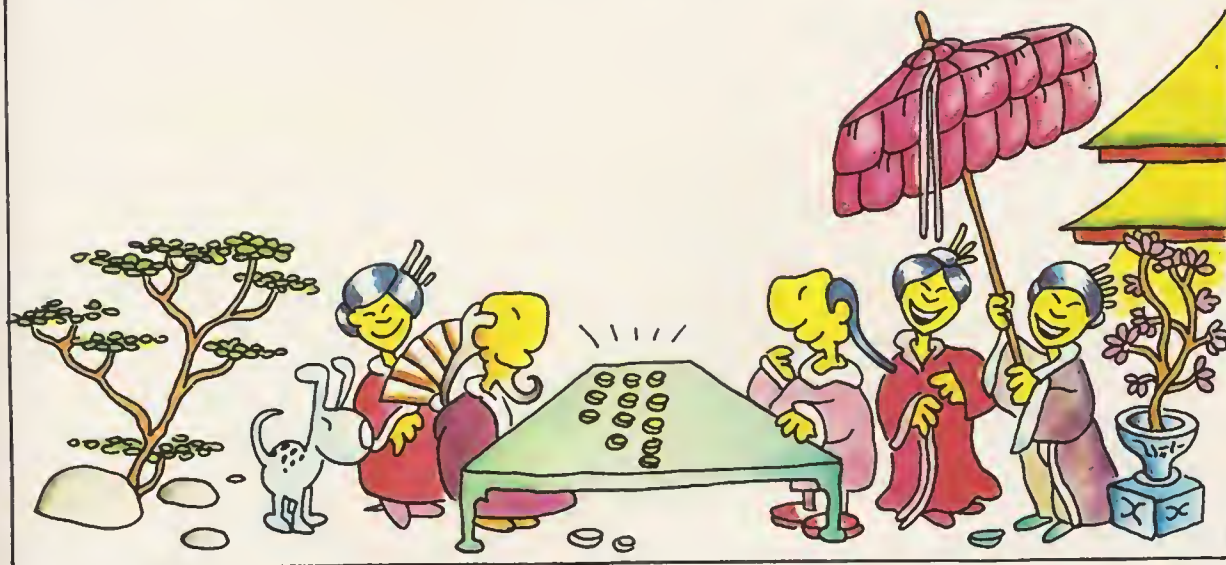
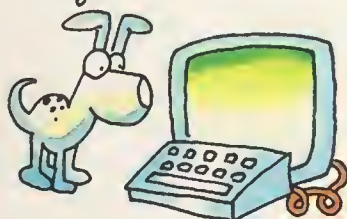
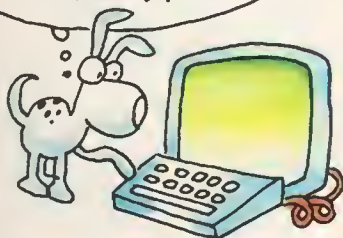
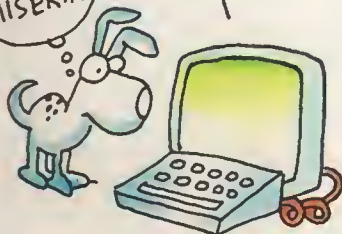
Gauss notò che la somma del primo numero con l'ultimo, 1 e 100, del secondo con il penultimo, 2 e 99, del terzo con il terzultimo, 3 e 98, e così via, dava sempre come risultato 101. Poichè queste coppie sono 50, moltiplicò 101 x 50 e risolse il problemino in un attimo, con grande costernazione del maestro antipatico.

FOR/TO/STEP/NEXT (step=gradino, passo). Il computer eseguirà ciclicamente le Istruzioni tra FOR/TO e NEXT finchè la "variabile di ciclo" non avrà raggiunto il valore indicato. L'incremento è di 1 salvo che sia indicato il passo (STEP). Nel nostro esempio è 2.

IL COMPUTER
E I NUMERI PRIMI

Con questo programmino il vostro computer troverà tutti i numeri primi compresi tra 3 ed N.

```
 10 REM NUMERI PRIMI
 20 CLS:PRINT"CERCO I NUMERI PRIMI COMPRESI TRA 3 E ?"
 30 INPUT N
 40 FOR K=3 TO N STEP 2
 50 P=0
 60 M=INT(K/2)
 70 REM CONSIDERO I POSSIBILI DIVISORI FRA 2 E K/2
 80 FOR J=2 TO M
 90 IF K/J=INT(K/J) THEN P=1
100 NEXT J
110 IF P=1 GOTO 130
120 PRINT K
130 NEXT K
140 END
```

FOR TO STEP NEXT
START
INPUTN
K=3
K>N
SI
NO
END
P=0
M=INT(K/2)
J=2
COSÌ E' STATO COSTRUITO IL PROGRAMMA PER LA RICERCA DEI NUMERI PRIMI.
J>M
SI
NO
P=1
SI
NO
K/J=INT(K/J)
SI
NO
P=1
J=J+1
PRINT K
K=K+1
UN PROGRAMMINO MOLTO SEMPLICE NO?
SEMPRE SPIRITOSI QUESTI PROGRAMMATORI!

GO SUB (=vai al sottoprogramma). I programmi più complessi possono essere composti da un programma principale e da più sottoprogrammi. Quando il computer incontra GO SUB lascia il programma principale e va nel sottoprogramma indicato. Per ritornare nel programma principale il computer deve incontrare l'ordine RETURN.

IL PIÙ ANTICO GIOCO DEL MONDO

Il gioco NIM è forse uno dei più antichi giochi conosciuti. Si gioca con dodici bastoncini o dodici pietruzze. Noi lo giocheremo come gli antichi cinesi, con dodici monete poste su tre file. I giocatori tolgono da una sola riga per volta una o più monete. Vince il giocatore che prende l'ultima.

```
10 REM IL GIOCO DEL NIM
20 DIM A(3)
30 A(1)=3
40 A(2)=4
50 A(3)=5
60 CLS
70 REM STAMPO LA CONFIGURAZIONE
80 GOSUB 380
90 PRINT:PRINT"DIMMI LA RIGA ED IL NUMERO DI MONETE"
100 INPUT F,N
110 REM CONTROLLO DATI
120 IF F>3 GOTO 170
130 IF N<=A(F) GOTO 200
140 IF A(F)=0 GOTO 170
150 PRINT:PRINT"IL NUMERO DELLE MONETE NON E' CORRETTO"
160 GOTO 90
170 PRINT:PRINT"LA RIGA NON E' VALIDA"
180 GOTO 90
190 REM SOTTRAGGO LE MONETE
200 GOSUB 490
210 IF C=1 GOTO 240
220 PRINT:PRINT"HAI VINTO TU"
230 GOTO 340
240 GOSUB 380
250 F=INT((3*RND)+1)
260 IF A(F)=0 GOTO 250
270 REM F E' UGUALE AD UNA DELLE TRE RIGHE
280 N=INT((A(F)*RND)+1)
290 PRINT:PRINT"ORA GIOCO IO. RIGA ";F;"MONETE ";N
300 GOSUB 500
310 IF C=0 GOTO 330
320 GOTO 80
330 PRINT:PRINT"HO VINTO IO"
340 PRINT"VUOI GIOCARE ANCORA ? SI/NO"
350 INPUT R$
360 IF R$="SI" GOTO 30
370 GOTO 570
380 REM SOTTOPROGRAMMA PER STAMPARE LE RIGHE
390 PRINT
400 FOR J=1 TO 3
410 PRINT J;" ";
420 IF A(J)=0 GOTO 460
430 FOR K=1 TO A(J)
440 PRINT"O ";
450 NEXT K
460 PRINT
470 NEXT J
480 RETURN
490 REM SOTTOPROGRAMMA PER SOTTRARRE LE MONETE
500 C=0
510 A(F)=A(F)-N
520 FOR K=1 TO 3
530 IF A(K)=0 GOTO 550
540 C=1
550 NEXT K
560 RETURN
570 END
```

GO TO (=vai a). Dice al computer di proseguire il programma alla linea indicata, che può essere avanti o indietro. È detta anche istruzione di salto incondizionato. Infatti quando il computer l'incontra da sola deve andare in qualsiasi caso alla riga indicata dopo GO TO.

EUCLIDE E IL MASSIMO COMUN DIVISORE

Euclide è l'inventore dell'algoritmo per il calcolo del M.C.D. di due numeri. È una "ricetta" che si può esprimere con le seguenti regole:

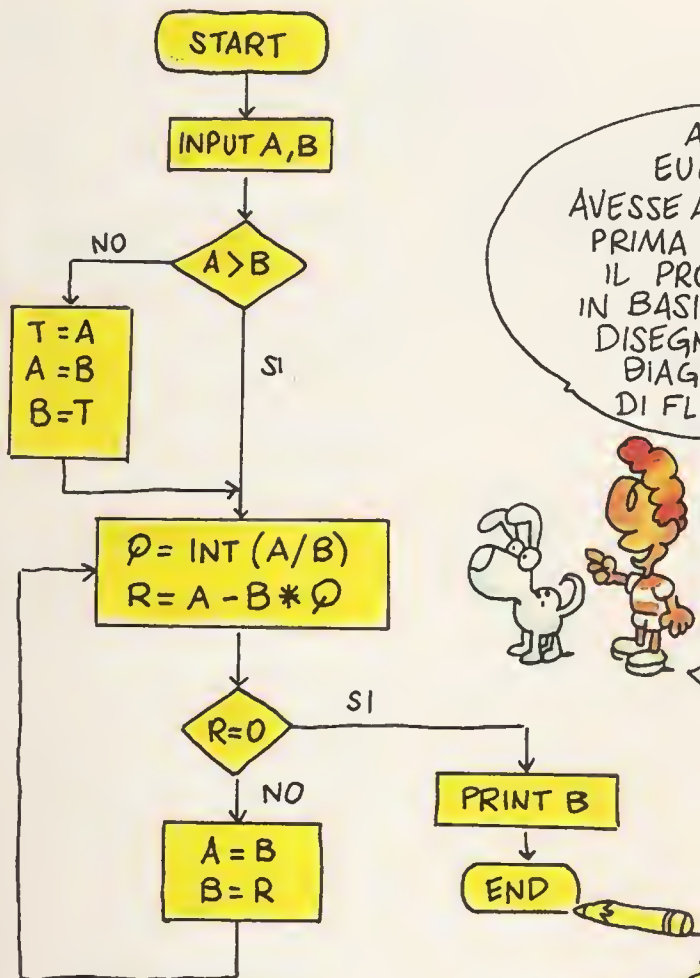
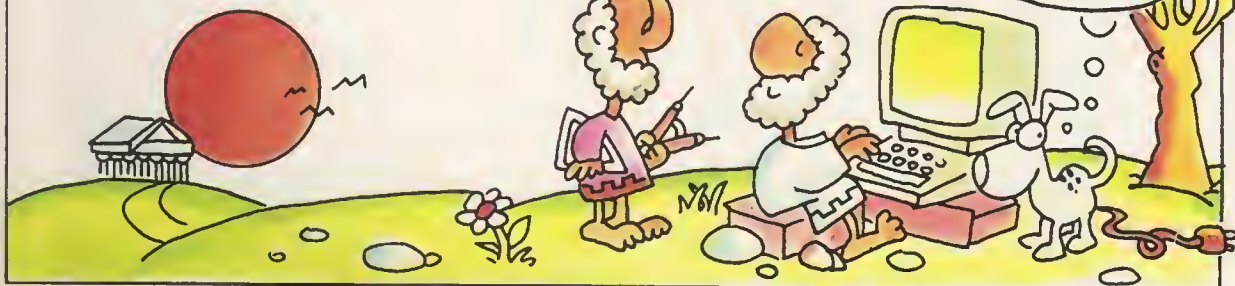
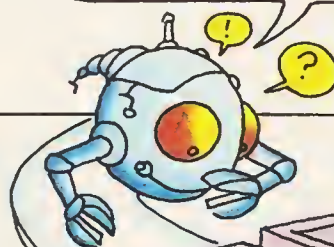
- se A e B sono i due numeri, con A maggiore di B, si divida A per B e sia R il resto
- se R è uguale a zero, B è il M.C.D.
- se R non è uguale a zero si sostituisca A con B e B con R ripetendo il procedimento da capo.

Ed ecco trasformate in istruzioni BASIC i vari passaggi dell'algoritmo di Euclide.

```
 10 REM CALCOLO DEL M.C.D.
 20 CLS:PRINT"IMMETTI I DUE NUMERI DI CUI VUOI"
 30 PRINT"CALCOLRE IL MASSIMO COMUN DIVISORE"
 40 INPUT A,B
 50 IF A>B GOTO 90
 60 T=A
 70 A=B
 80 B=T
 90 Q=INT(A/B)
100 R=A-B*Q
110 IF R=0 GOTO 150
120 A=B
130 B=R
140 GOTO 90
150 PRINT:PRINT"MASSIMO COMUN DIVISORE = ";B
160 END
```

L'istruzione 30 si chiede se A è maggiore di B: se si, salta all'istruzione 70, altrimenti esegue le istruzioni di seguito dove, usando un campo di comodo T, si scambiano fra di loro i valori di A e B. Nei BASIC più avanzati è possibile scrivere più di una assegnazione su di una stessa riga, separandole con ":", pertanto invece delle 40-50-60 si poteva scrivere:
40 T=A:A=B:N=T

GO TO

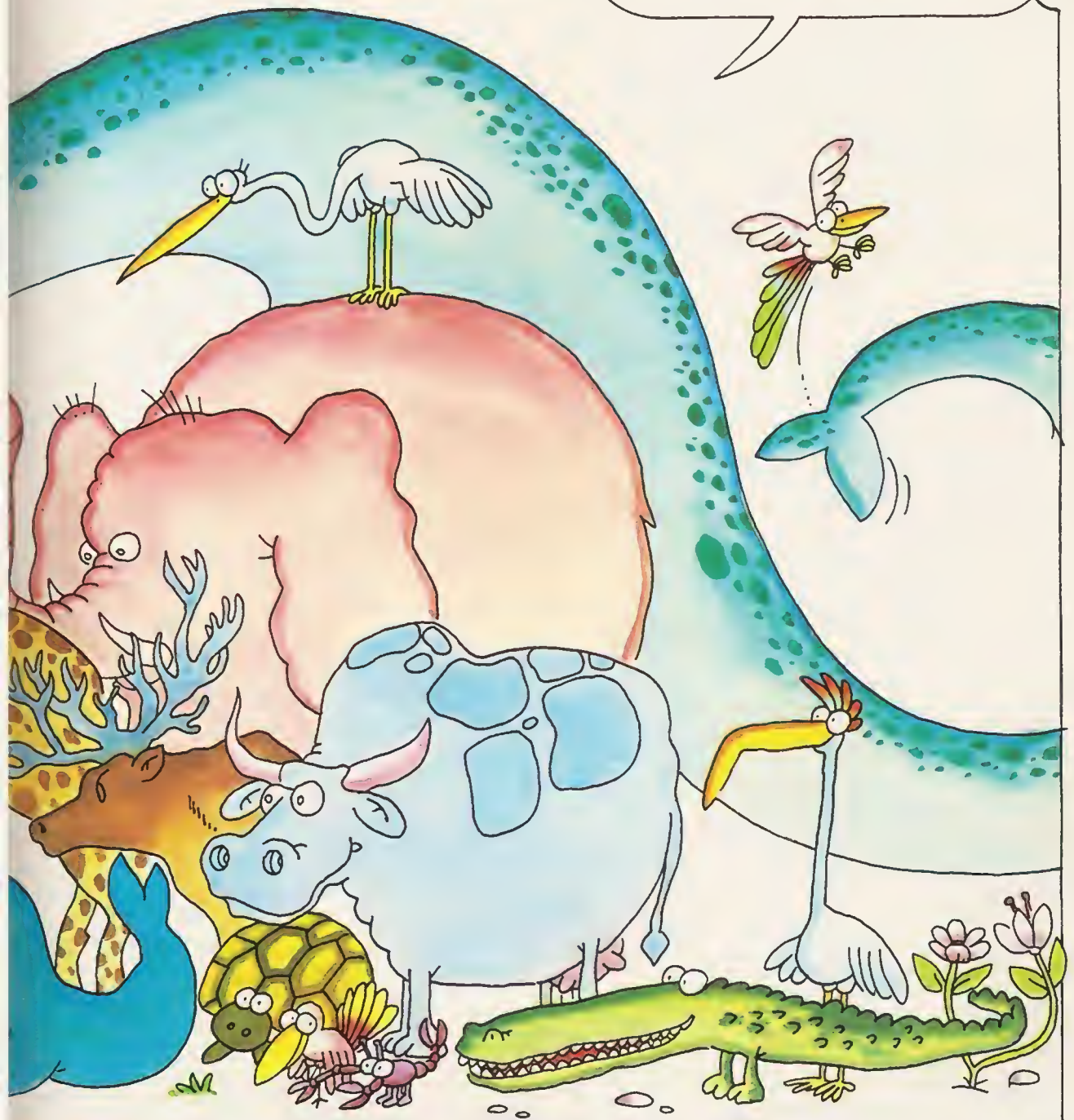
Le istruzioni 70-90 calcolano il quoziente ed il resto della divisione di A e B. Se il resto è uguale a zero il procedimento è terminato e si salta alla istruzione di PRINT, altrimenti si pone A=B B=R e si ricomincia dalla 70, come prescrivono le regole dell'algoritmo.

IF (=se). Mette il computer davanti a una scelta che dipende da una condizione. Se questa si verifica, allora (THEN) fa qualcosa oppure va a (GO TO) un'altra riga. Nel nostro esempio quando A$="SI" allora il computer "salta" a tre righe più avanti e scrive l'animale che avete pensato.

IL COMPUTER INDOVINO

Con questo programmino potete "insegnare" al vostro computer a indovinare un animale che avete pensato. Attenzione, guardate bene come è costruito. Abbiamo messo anche "un ciclo d'attesa" che non è altro che un ciclo FOR/TO/NEXT a vuoto, fatto solo per non aspettare il computer.

```
 10 REM IL COMPUTER INDOVINO
 20 CLS: PRINT"PENSA AD UN ANIMALE"
 30 PRINT"TRA QUESTI TRE"
 40 PRINT"RAGNO  TIGRE  SERPENTE"
 50 FOR I=1 TO 3000 :NEXT I
 60 PRINT"HAI PENSATO ?"
 70 INPUT R$
 80 IF NOT R$="SI" THEN GOTO 50
 90 PRINT"RISPONDI ORA ALLE SEGUENTI DOMANDE"
100 PRINT"FA LE RAGNATELE ?"
110 INPUT A$
120 IF A$="SI" THEN GOTO 170
130 PRINT"E' UN MAMMIFERO ?"
140 INPUT A$
150 IF A$="SI" THEN GOTO 180
160 PRINT"HAI PENSATO AL SERPENTE":GOTO 190
170 PRINT"HAI PENSATO AL RAGNO":GOTO 190
180 PRINT"HAI PENSATO ALLA TIGRE":GOTO 190
190 END
```

FATELO DA VOI

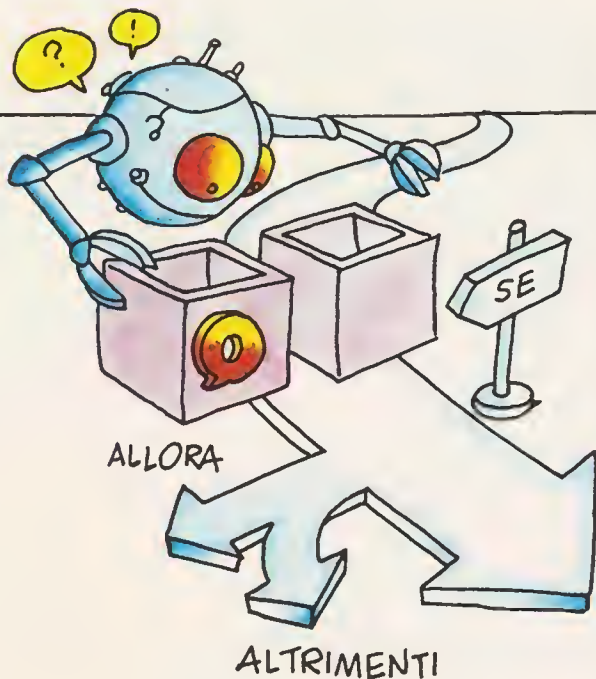
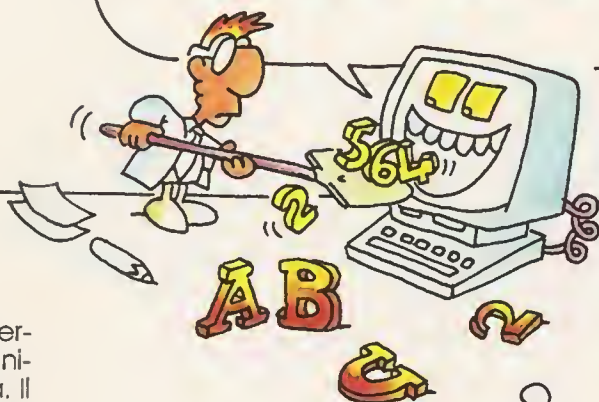
Se avete capito come funziona il programmino della pagina precedente potrete facilmente cambiare gli animali che il computer può "indovinare". Potrete anche allungare il programma "insegnandogli" altri animali, anche tutti quelli disegnati in questa pagina. Per renderlo più misterioso vi consigliamo di premettere all'inizio del programma e prima di ogni domanda una riga con l'ordine CLS (o quello corrispondente sul vostro computer). Così il video si pulirà ogni volta e sembrerà che il computer dialoghi con i vostri amici. Sarà insomma il primo videogioco costruito tutto da voi. Buon divertimento.

IF GO TO THEN

IF GOTO IF THEN ELSE

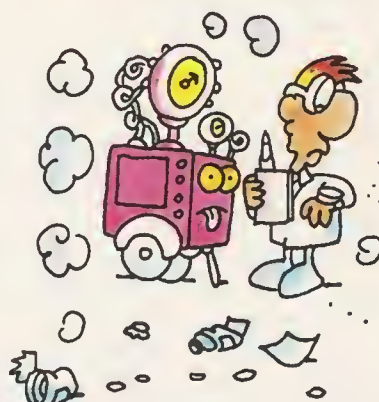
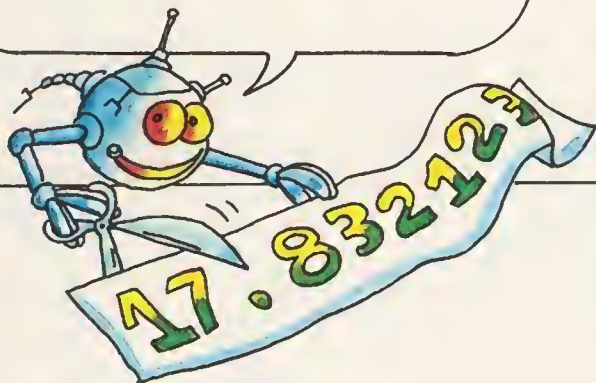
ELSE (=altrimenti). Permette al computer di fare due cose dopo una scelta: se (IF) la condizione si verifica, allora (THEN) fa una cosa, altrimenti (ELSE) ne fa un'altra. Nel nostro caso se Q=3 allora deve andare alla riga 290, altrimenti alla riga 100.

UN ALTRO GIOCO

Il computer "pensa" un numero di tre cifre e voi dovete indovinarlo. Coraggio, forse vi aiuterà.

```
 10 REM INDOVINA UN NUMERO DI TRE CIFRE
 20 DIM M(3),T(3),B$(3)
 30 CLS
 40 REM NUMERO DI GIOCATE TRA 3 E 10
 50 G=INT((RND*7)+3)
 60 C=0
 70 REM TRE CIFRE A CASO TRA 0 E 9
 80 FOR K=1 TO 3
 90 M(K)=INT(RND*10)
100 NEXT K
110 C=C+1
120 IF C>G THEN 320
130 PRINT:PRINT"GIOCATA N.   :";C
140 PRINT"IMMETTI LE TUE CIFRE"
150 INPUT T(1),T(2),T(3)
160 REM CONTROLLO DI ESATTEZZA
170 B$(1)="NO":B$(2)="NO":B$(3)="NO":Q=0
180 FOR P=1 TO 3
190 FOR K=1 TO 3
200 IF M(K)=T(P) AND K=P THEN 230
210 IF M(K)=T(P) AND K<>P THEN B$(P)="NI"
220 GOTO 240
230 Q=Q+1:B$(P)="SI":GOTO 250
240 NEXT K
250 NEXT P
260 REM EVIDENZIO IL RISULTATO
270 PRINT "  ";B$(1);B$(2):B$(3)
280 REM CONTROLLO IL NUMERO DI CIFRE CORRETTE
290 IF Q=3 THEN 300 ELSE 110
300 PRINT"HAI INDOVINATO IL NUMERO"
310 GOTO 340
320 PRINT"NON HAI INDOVINATO ENTRO IL LIMITE DI ";G;" GIOCATE"
330 PRINT"IL MIO NUMERO ERA:  ";M(1);M(2);M(3)
340 PRINT"VUOI GIOCARE ANCORA ? SI/NO"
350 INPUT S$
360 IF S$="SI" THEN 30
370 END
```

INPUT (=ingresso). Dice al computer di chiedere dati dall'esterno. Quando incontra questa istruzione si ferma e aspetta che voi scriviate con la tastiera i numeri o le parole di cui ha bisogno. Nel nostro esempio aspetta i tassi di anidride solforosa e l'ora dei rilevamenti.

PROFESSIONISTI DELL'INQUINAMENTO

L'inquinamento si misura rilevando a intervalli più o meno regolari la quantità di anidride solforosa presente nell'atmosfera. Il problema è di calcolare la media su ogni singolo intervallo di tempo e la media ponderata sulle medie.
Ecco il programmino che risolve il problema:

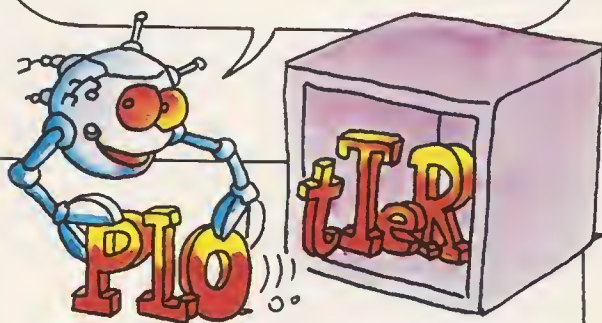
```
10 REM INQUINAMENTO
20 CLS:PRINT"QUANTI RILEVAMENTI HAI FATTO ?"
30 INPUT N
40 REM R=VALORI RILEVAMENTI T=TEMPI
50 DIM R(N),T(N)
60 PRINT"IMMETTI I VALORI E L' ORA DEI RILEVAMENTI :"
70 FOR K=1 TO N
80 INPUT R(K),T(K)
90 NEXT K
100 S=0:PRINT
110 FOR K=1 TO N-1
120 REM MEDIA
130 M=(R(K)+R(K+1))/2
140 PRINT"MEDIA INTERVALLO ";T(K);T(K+1);" = ";M
150 I=T(K+1)-T(K)
160 S=S+M*I
170 NEXT K
180 P=S/(T(N)-T(1))
190 PRINT:PRINT"MEDIA PONDERATA :";P
200 END
```

INT (da INTeger=intero). Dice al computer di calcolare solo la parte intera di un numero, quella prima del punto. Per esempio INT(57.79) è 57. Attenzione che nel caso di numeri negativi INT calcola il valore intero immediatamente inferiore. Per esempio INT(-5.3) è -6.

MARKO E IL MINIMO COMUNE MULTIPLO

Marko a scuola deve spesso calcolare il minimo comune multiplo di due numeri. È una cosa che odia. Perciò ha costruito un programmino in modo che il computer lo calcoli al posto suo. Per costruirlo ha usato questo algoritmo:

- se A e B sono due numeri con A maggiore di B, se A è divisibile per B allora A è il m.c.m.
- altrimenti troviamo il più piccolo multiplo di B che sia maggiore di A
- se questo è anche multiplo di A abbiamo trovato il m.c.m.
- altrimenti cerchiamo il successivo multiplo di B ripetendo il procedimento.

```
 10 REM CALCOLO DEL MINIMO COMUNE MULTIPLO
 20 CLS:PRINT"IMMETTI I DUE NUMERI"
 30 INPUT A,B
 40 IF A>B GOTO 60
 50 T=A:A=B:B=T
 60 IF A/B=INT(A/B) GOTO 130
 70 D=INT(A/B)
 80 D=D+1
 90 M=D*B
100 R=M-A*INT(M/A)
110 IF R<>0 GOTO 80
120 A=M
130 PRINT"MINIMO COMUNE MULTIPLO:  ";A
140 END
```

L'istruzione 40 scambia i valori di A e di B qualora B fosse maggiore di A.

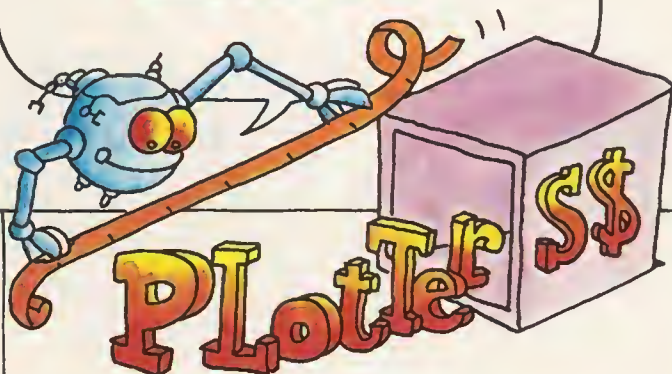
LEFT$ (left=sinistra). Dice al computer di estrarre la parte più a sinistra di una stringa di caratteri. Se A$="Plotter", LEFT$ (A$,3) è "PLO". Nello Spectrum LEFT$ va omesso, dovremmo scrivere A$ (1 TO 3). Nell'esempio sottostante LEFT$ serve per estrarre la prima cifra a sinistra del numero che volete "tradurre".

I NUMERI ROMANI

Ecco il programma che traduce i numeri compresi tra 1 e 1000 nelle cifre che usavano gli antichi romani.
Ricordiamo che:

I=1 II=2 III=3 IV=4 V=5 VI=6 VII=7 VIII=8 IX=9 X=10 XL=40 L=50 LX=60 XC=90 C=100 CX=110 CL=150 CLX=160 D=500 DL=550 M=1000

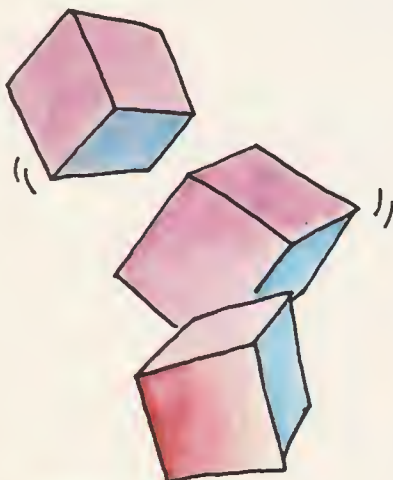
```
 10 REM CONVERTIAMO IN NUMERO ROMANO
 20 DIM V$(3,9)
 30 DATA I,II,III,IV,V,VI,VII,VIII,IX
 40 DATA X,XX,XXX,XL,L,LX,LXX,LXXX,XC
 50 DATA C,CC,CCC,CD,D,DC,DCC,DCCC,CM
 60 FOR I=1 TO 3
 70 FOR J=1 TO 9
 80 READ V$(I,J)
 90 NEXT J
100 NEXT I
110 CLS:PRINT"IMMETTI UN NUMERO DECIMALE"
120 PRINT"INFERIORE O UGUALE A 1000"
130 INPUT N
140 IF N>1000 GOTO 120
150 N$=RIGHT$(STR$(N),LEN(STR$(N))-1)
160 L=LEN(N$)
170 IF L=4 GOTO 280
180 R$=" "
190 FOR K=L TO 1 STEP -1
200 Q$=LEFT$(N$,1)
210 Q=VAL(Q$)
220 IF Q=0 GOTO 240
230 R$=R$+V$(K,Q)
240 N$=RIGHT$(N$,K-1)
250 NEXT K
260 GOTO 290
270 REM R$=M QUANDO N=1000
280 R$="M"
290 PRINT:PRINT"IL NUMERO DECIMALE :   ";N
300 PRINT:PRINT"IN NUMERO ROMANO SI ESPRIME :   ";R$
310 END
```

LEN

LEN (da LENght=lunghezza). Dice al computer di contare quanti caratteri sono contenuti in una stringa. Conta anche gli spazi vuoti. LEN ("W PLOTTER") è 9.
Nel nostro esempio LEN è usato alla riga 40.

E ADESSO INVENTIAMO LE PAROLE

Scriviamo una "stringa" di caratteri qualsiasi S$. La faremo invertire dal computer che ci dirà anche se è uguale a se stessa leggendola da sinistra o da destra.

```
 10 CLS:PRINT"IMMETTI UNA STRINGA DI CARATTERI"
 20 INPUT S$
 30 A$=S$
 40 L=LEN(S$)
 50 FOR K=1 TO L
 60 A$=LEFT$(A$,L-K+1)
 70 R$=RIGHT$(A$,1)
 80 B$=B$+R$
 90 NEXT K
100 PRINT:PRINT"STRINGA IMMESSA    ";S$
110 PRINT:PRINT"STRINGA RIBALTATA  ";B$
120 IF B$<>S$ GOTO 150
130 PRINT:PRINT"LA STRIGA E' IDENTICA NEI DUE SENSI"
140 GOTO 160
150 PRINT:PRINT"LA STRINGA NON E' IDENTICA NEI DUE SENSI"
160 END
```

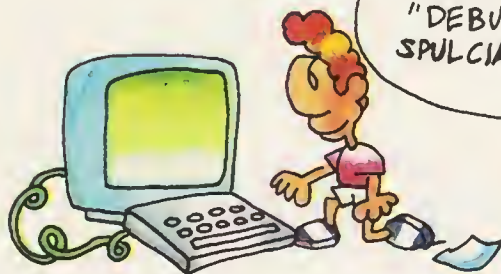
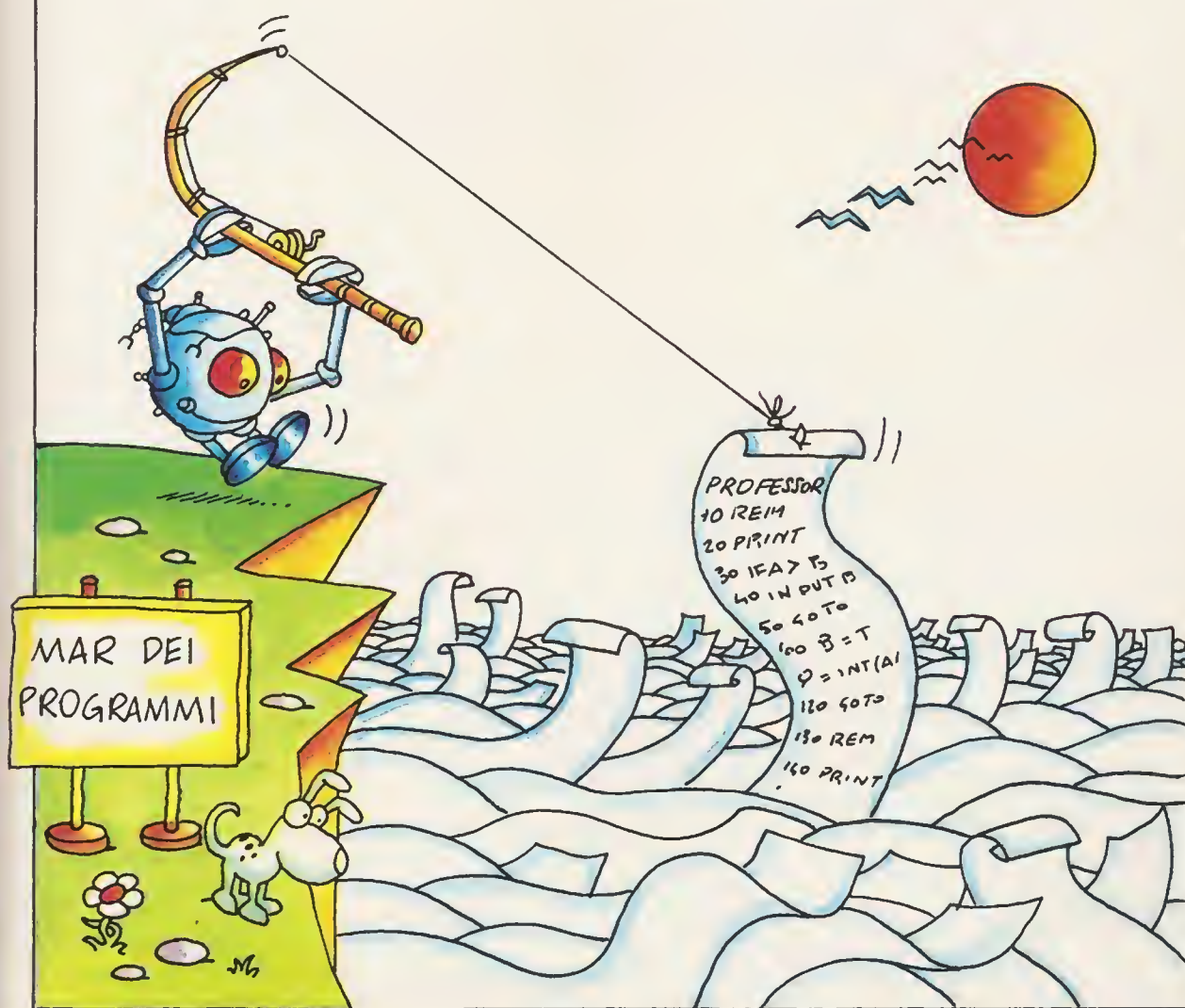
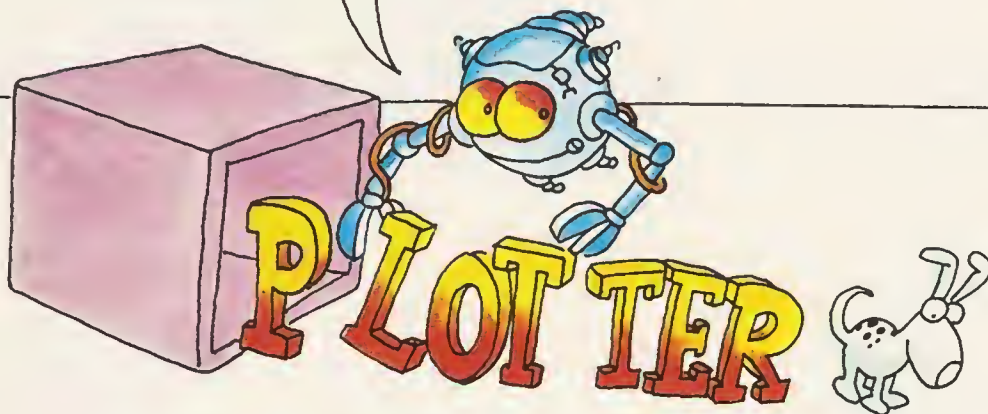
L'istruzione 40 calcola la lunghezza della stringa S$, questo valore L è poi usato nella iterazione come limite superiore per K.
L'istruzione 60 considera la stringa d'appoggio A$ e ne estrae (L—K+1) elementi più a sinistra; la 70 mette in R$ l'elemento più a destra di A$.

LET (=metti che, poni). Dice al computer di assegnare un valore ad una variabile (per esempio LET A=5). Ma si può usare per assegnare ad una variabile anche un'intera espressione, come nel nostro esempio. LET può essere omesso, il computer capirà ugualmente.

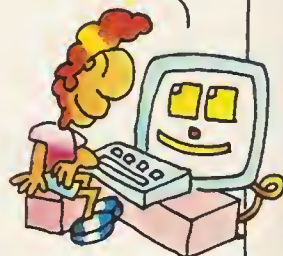
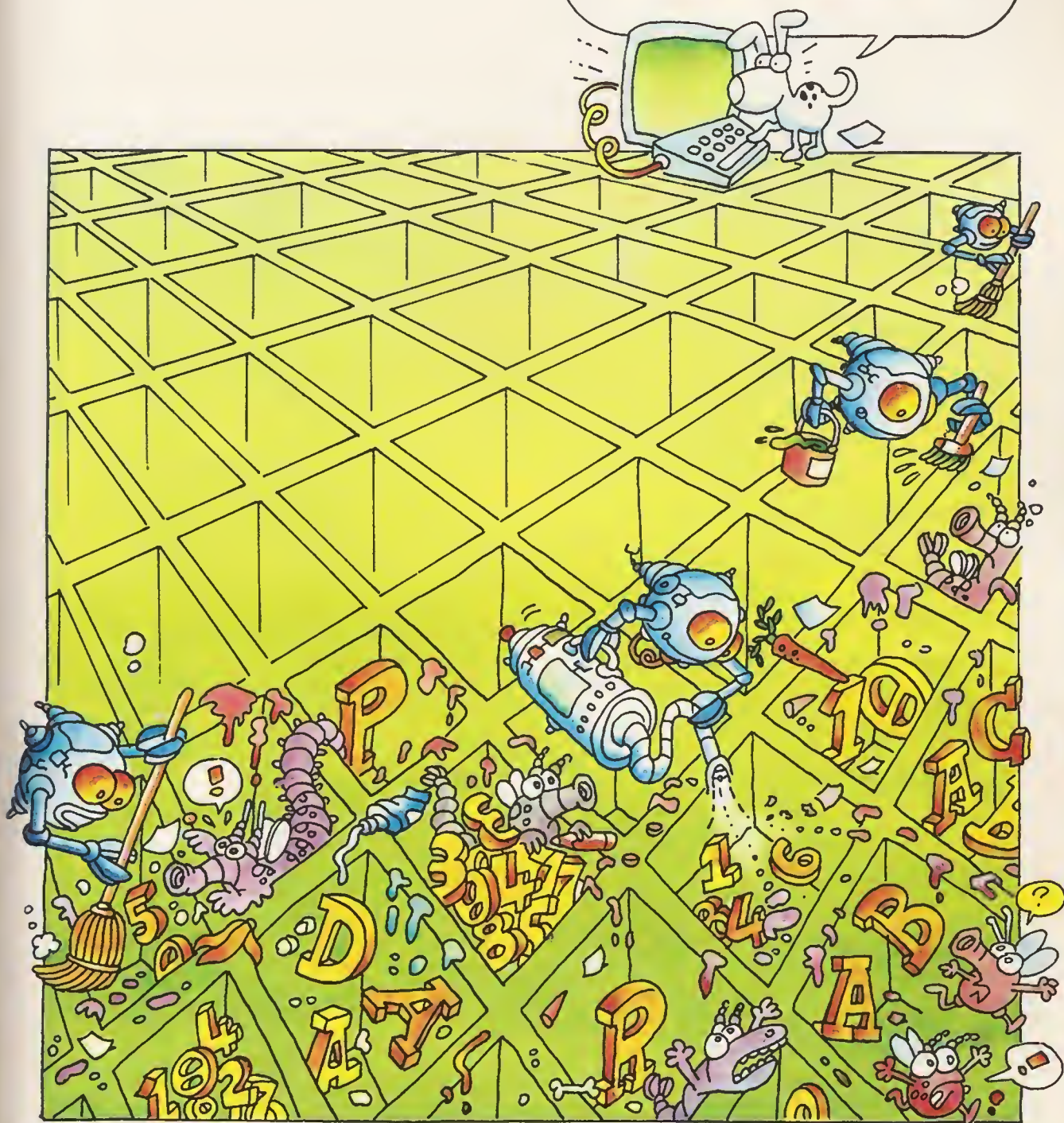
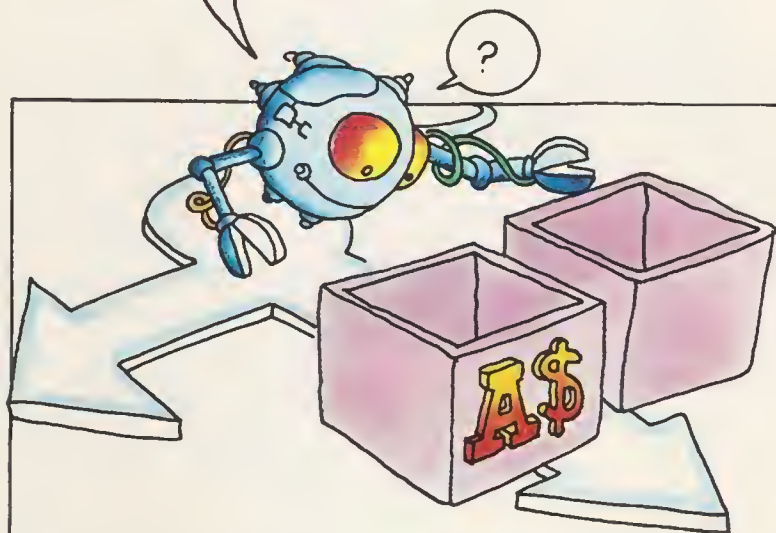
CHI ARRIVERÀ PRIMA?

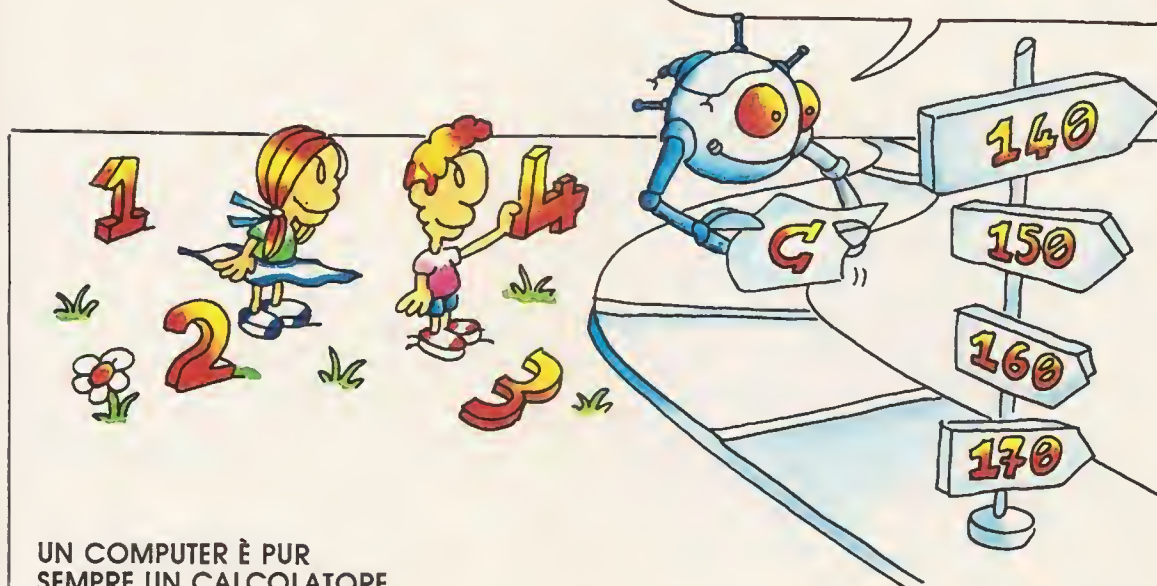
Ada e Marko vanno in vacanza insieme, ma partono su due auto diverse, Marko sull'auto N° 1 e Ada sulla N° 2. Chi arriverà prima? Si accettano scommesse.

```
 10 REM CHI ARRIVERA' PRIMA?
 20 CLS
 30 PRINT:PRINT"IMMETTI LA DISTANZA DA PERCORRERE :"
 40 INPUT D
 50 PRINT:PRINT"IMMETTI LE VELOCITA' :"
 60 INPUT V1,V2
 70 PRINT:PRINT"IMMETTI GLI ORARI DELLE PARTENZE :"
 80 INPUT T1,T2
 90 LET R=D/V1+T1
100 LET S=D/V2+T2
110 IF R<S GOTO 150
120 IF R>S GOTO 170
130 PRINT:PRINT"ARRIVANO INSIEME"
140 GOTO 180
150 PRINT:PRINT"L' AUTO NO. 1 ARRIVA PRIMA"
160 GOTO 180
170 PRINT:PRINT"L' AUTO NO. 2 ARRIVA PRIMA"
180 END
```

LIST

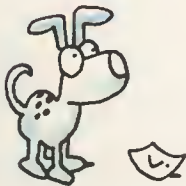
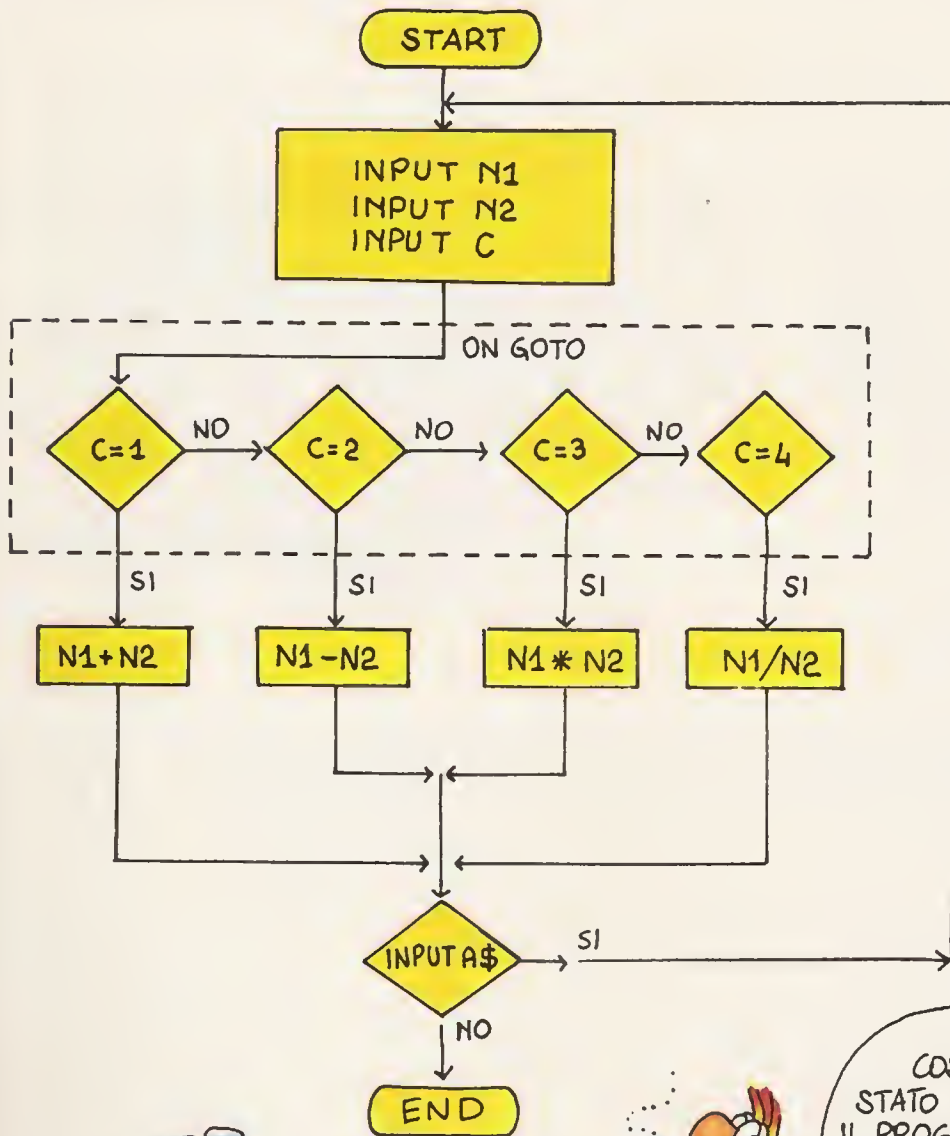
LIST (=fare una lista, "listare"). Ordina al computer di mostrare il programma. Da solo LIST visualizza tutto il programma. Seguito dal numero di riga fa apparire la riga indicata. Seguito da due righe visualizza la "lista" di righe compresa tra le due.

LOAD (=caricare). Ordina al computer di caricare nella propria memoria un programma che avevate precedentemente "salvato" su cassetta o dischetto magnetico. Dopo LOAD occorre mettere il nome del programma in modo che il computer possa riconoscerlo e ripescarlo tra tanti altri. Per esempio LOAD "PROFESSOR".

MID$ (da MIDdle=nel mezzo). Permette di estrarre uno o più caratteri contenuti in una stringa. Per esempio MID$ ("PLOTTER",2,3) è LOT. Nello Spectrum dovremmo omettere MID$ e scrivere solo PLOTTER (2 TO 4). Nel nostro esercizio MID$ serve per estrarre le prime lettere dei nomi immessi nel programma per poi confrontarle e ordinare i nomi.

```
 10 REM PROGRAMMA DI ORDINAMENTO
 20 CLS:PRINT"IMMETTI IL NUMERO DI NOMI DA ORDINARE (MAX 30)"
 30 INPUT NM
 40 IF NM<1 OR NM>30 GOTO 20
 50 DIM MN$(NM)
 60 FOR I=1 TO NM
 70 PRINT"IMMETTI IL NOME N.RO";I;"(MAX 24 CARATTERI)"
 80 INPUT MN$(I)
 90 IF LEN(MN$(I))>24 GOTO 70
100 NEXT I
110 FL=0
120 FOR I=1 TO NM - 1
130 FOR J=1 TO 29
140 A$=MID$(MN$(I),J,J):B$=MID$(MN$(I+1),J,J)
150 IF A$="" OR B$="" GOTO 200
160 A=ASC(A$):B=ASC(B$)
170 IF B>A GOTO 200
180 IF A=B THEN NEXT J
190 IF B<A THEN GOSUB 280
200 NEXT I
210 IF FL=1 GOTO 110
220 CLS
230 PRINT" *--- NOMI ORDINATI ALFABETICAMENTE ---*":PRINT
240 FOR I=1 TO NM
250 PRINT "        ";MN$(I)
260 NEXT I
270 END

280 S$=MN$(I)
290 MN$(I)=MN$(I+1)
300 MN$(I+1)=S$
310 FL=1
320 RETURN
```

ADA
ALBERTA
ALDO
AUGUSTO
BETTY...

NEW (=nuovo). Ordina al computer di cancellare dalla memoria ogni traccia dei programmi precedenti. È consigliabile usare NEW sempre prima di cominciare a scrivere un nuovo programma.

NOT (=no). Cambia il valore di una variabile logica. Per esempio se A è vero, NOT A è falso e viceversa. Nel nostro programma-quiz lo troviamo insieme a AND. Solo se (IF) la prima condizione si verifica e (AND) non (NOT) la seconda, il computer andrà alla riga indicata, altrimenti proseguirà alla riga successiva.

```
 10 REM QUIZ SUL BASIC
 20 DIM MD$(8,2)
 30 DATA "UNA STRINGA E' UNA SEQUENZA DI CARATTERI TRA VIRGOLETTE ?","SI"
 40 DATA "LIST, SAVE, LOAD, NEW VOGLIONO IL NUMERO DI RIGA ?","NO"
 50 DATA "QUAL' E'L ISTRUZIONE CHE SERVE PER RIVEDERE UN PROGRAMMA ?","LIST"
 60 DATA "NEW E' L' ORDINE CHE SGOMBRA TUTTA LA MEMORIA ?","SI"
 70 DATA "QUAL' E' L'ISTRUZIONE CHE GENERA UN NUMERO A CASO ?","RND"
 80 DATA "QUAL' E' L' ISTRUZIONE CHE SCRIVE DIRETTAMENTE NELLA MEMORIA ?","POKE"
 90 DATA "QUANTI BYTE FORMANO UN K-BITE ?","1024"
100 DATA "SAVE E' UN' ISTRUZIONE BASIC ?","SI"
110 FOR I=1 TO 8
120 READ MD$(I,1),MD$(I,2)
130 NEXT I
140 D1=INT(RND*8+1)
150 D2=INT(RND*8+1)
160 IF D1=D2 GOTO 150
170 CLS:PRINT"RISPONDI ALLE SEGUENTI DOMANDE"
180 PRINT:PRINT MD$(D1,1)
190 PRINT:PRINT MD$(D2,1)
200 PRINT:INPUT R1$,R2$
210 IF R1$=MD$(D1,2) AND R2$=MD$(D2,2) GOTO 250
220 IF R1$=MD$(D1,2) AND NOT R2$=MD$(D2,2)GOTO 260
230 IF NOT R1$=MD$(D1,2) AND R2$=MD$(D2,2)GOTO 270
240 IF NOT R1$=MD$(D1,2) AND NOT R2$=MD$(D2,2)GOTO 280
250 PRINT:PRINT"BRAVO ! HAI INDOVINATO":GOTO 290
260 PRINT:PRINT"HAI INDOVINATO SOLO LA PRIMA. RIPROVA":GOTO 180
270 PRINT:PRINT"HAI INDOVINATO SOLO LA SECONDA. RIPROVA":GOTO 180
280 PRINT:PRINT"SONO SBAGLIATE ENTRAMBE. RIPROVA":GOTO 180
290 PRINT:PRINT"VUOI RISPONDERE AD ALTRE DOMANDE ?"
300 INPUT A$
310 IF A$="SI"GOTO 140
320 END
```

ON GO TO (= nel caso che, vai a). Dice al computer di "saltare" ad una delle righe indicate in funzione di un parametro. Nel nostro esempio il parametro è C. Se C=1 il computer prosegue alla riga 140, se C=2 prosegue alla riga 150, se C=3 prosegue alla riga 160, se C=4 alla riga 170.

UN COMPUTER È PUR SEMPRE UN CALCOLATORE

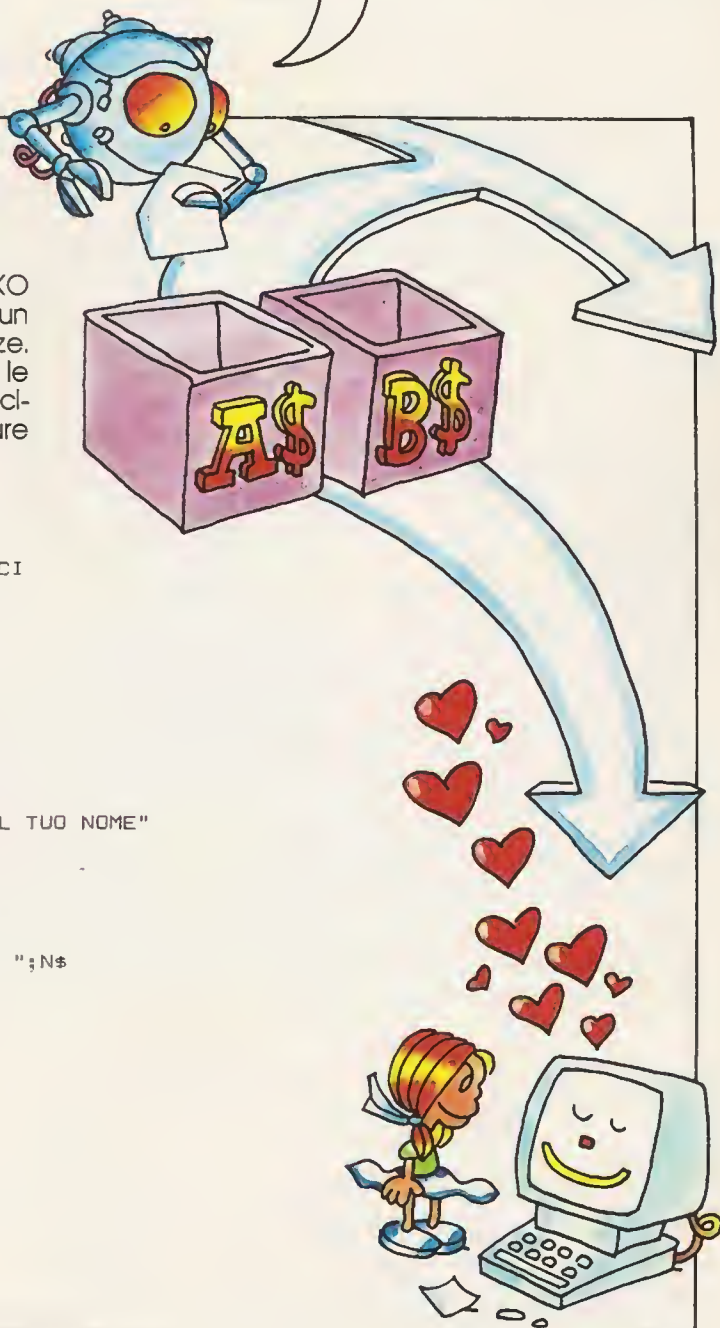
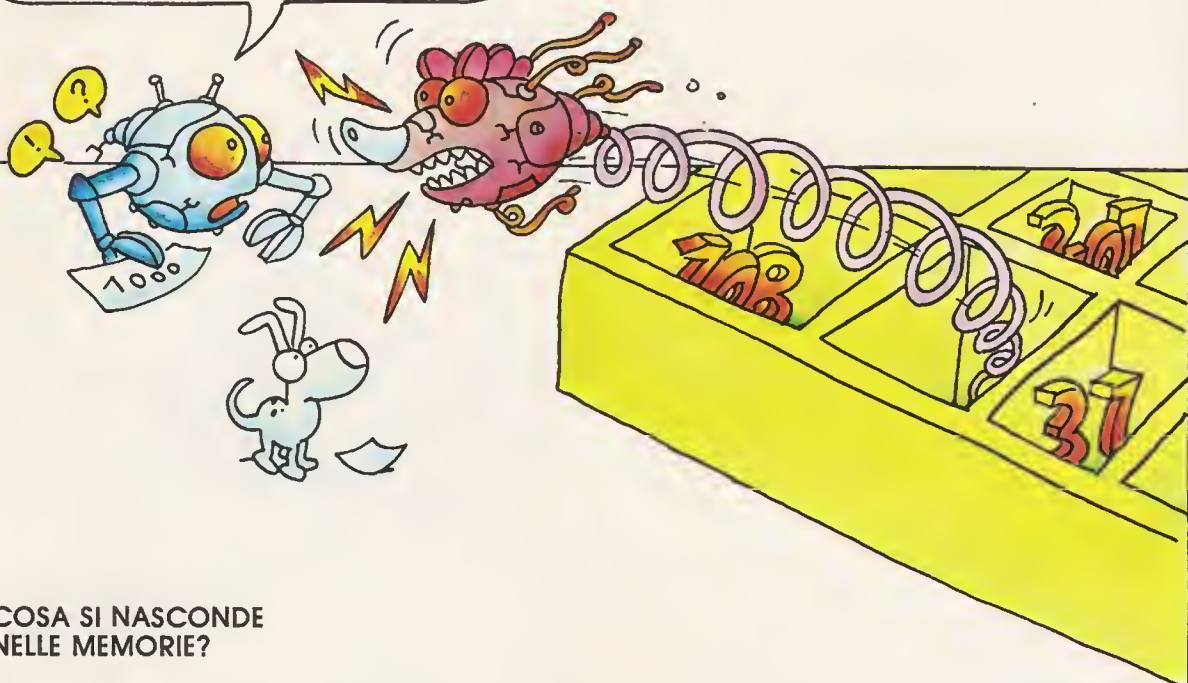
Fate "girare" questo programma. Inserite due numeri e in funzione della vostra scelta il computer calcolerà la somma, la sottrazione, la moltiplicazione o la divisione.

```
 10 REM LA CALCOLATRICE
 20 CLS:PRINT"IMMETTI IL PRIMO NUMERO"
 30 INPUT N1
 40 PRINT:PRINT"IMMETTI IL SECONDO NUMERO"
 50 INPUT N2
 60 CLS:PRINT"CHE OPERAZIONE DESIDERI ?":PRINT
 70 PRINT"         PREMI  1  PER L' ADDIZIONE":PRINT
 80 PRINT"         PREMI  2  PER LA SOTTRAZIONE":PRINT
 90 PRINT"         PREMI  3  PER LA MOLTIPLICAZIONE":PRINT
100 PRINT"         PREMI  4  PER LA DIVISIONE":PRINT
110 INPUT C
120 PRINT
130 ON C GOTO 150,160,170,180
140 REM ESEGUO I CALCOLI
150 PRINT N1;" + ";N2;" = ";N1+N2:GOTO 190
160 PRINT N1;" - ";N2;" = ";N1-N2:GOTO 190
170 PRINT N1;" * ";N2;" = ";N1*N2:GOTO 190
180 PRINT N1;" : ";N2;" = ";N1/N2:GOTO 190
190 PRINT:PRINT"VUOI FARE UN' ALTRA OPERAZIONE ?"
200 INPUT A$
210 IF A$="SI" GOTO 10
220 END
```

ON GOTO
START
INPUT N1
INPUT N2
INPUT C
ON GOTO
C=1
NO
C=2
NO
C=3
NO
C=4
SI
SI
SI
SI
N1+N2
N1-N2
N1*N2
N1/N2
INPUT A$
SI
NO
END
COSÌ E' STATO COSTRUITO IL PROGRAMMA DELLE QUATTRO OPERAZIONI.
CHIARO, NO?

OR (=o). È una "congiunzione" anche in BASIC. Se (IF) una condizione si verifica o (OR) un'altra, il computer allora (THEN) fa qualcosa, altrimenti prosegue alla riga succesiva. In conclusione è sufficiente che solo una delle condizioni poste sia vera per far eseguire al computer l'istruzione contenuta nella riga.

UN DISTRIBUTORE AUTOMATICO DI COCCOLE

Con questo programmino ADA e MARKO hanno trasformato il loro computer in un distributore automatico di baci e carezze. OR è usato nelle righe che controllano le risposte e nella riga dove il computer decide se "distribuire" baci e carezze oppure no.

```
 10 REM BACI E CAREZZE SOLO AGLI AMICI
 20 PRINT"TI PIACE PLOTTER ?"
 30 PRINT"RISPONDI SI NO"
 40 INPUT A$
 50 IF A$="SI" OR A$="NO" GOTO 70
 60 GOTO 30
 70 PRINT"PREFERISCI ADA E MARCO ?"
 80 PRINT"RISPONDI SI O NO"
 90 INPUT B$
100 IF B$="SI" OR B$="NO" GOTO 120
110 GOTO 80
120 PRINT"COME TI CHIAMI ?  SCRIVI IL TUO NOME"
130 INPUT N$
140 IF A$="SI" OR B$="SI" GOTO 160
150 PRINT"NON MI PIACI ";N$:END
160 PRINT"1 BACIO E 1 CAREZZA A ";N$
170 FOR K=2 TO 20
180 PRINT K;" BACI E ";K;" CAREZZE A ";N$
190 NEXT K
200 PRINT"SPERO CHE TI BASTINO ";N$
210 END
```

PEEK(to peek=vedere attraverso una fessura). Permette di leggere direttamente il contenuto di una cella di memoria. Se per esempio con PRINT PEEK (1000) otteniamo 123 vuol dire che nella cella di memoria N° 1000 c'è il valore 123 che può rappresentare un dato o una istruzione.

COSA SI NASCONDE NELLE MEMORIE?

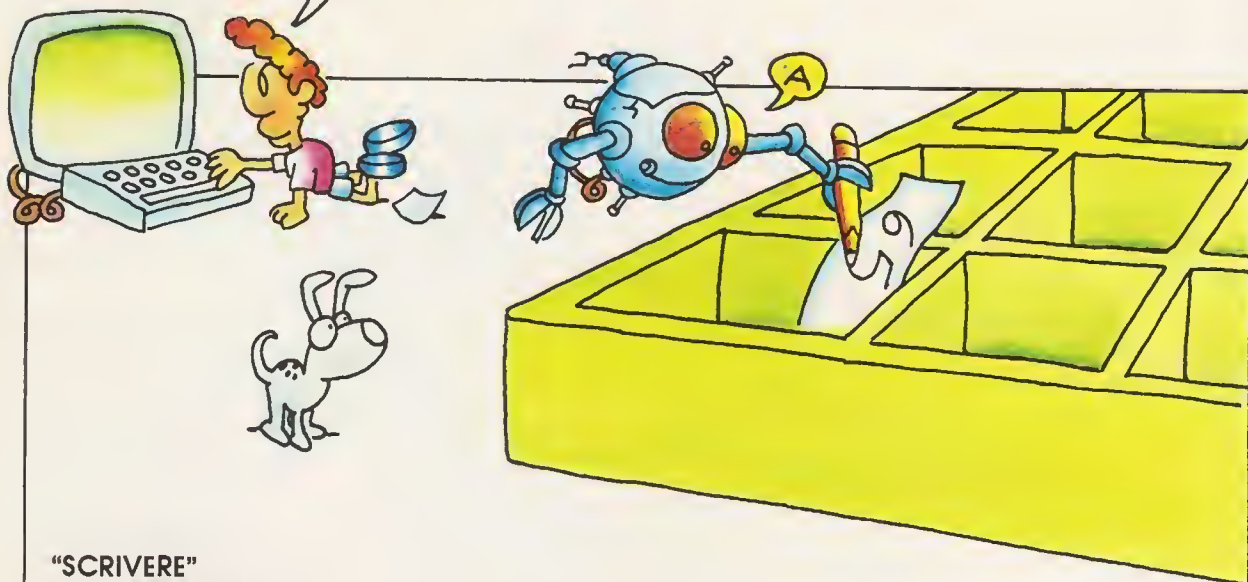
Con questo programmino potete esplorare le memorie del vostro computer. Troverete celle vuote (valore 0) oppure valori compresi tra 1 e 255. Questi valori esprimono lettere, simboli grafici, numeri o istruzioni in "linguaggio macchina".

```
 10 REM LETTURA DELLA MEMORIA
 20 CLS:PRINT"QUESTO PROGRAMMA PERMETTE DI LEGGERE"
 30 PRINT"LA MEMORIA DEL CALCOLATORE":PRINT
 40 PRINT"DI QUANTI K-BYTE E' LA MEMORIA DEL TUO CALCOLATORE?"
 50 INPUT KM
 60 M=KM*1024-1
 70 CLS:PRINT:PRINT"IMMETTI L' INDIRIZZO INIZIALE TRA O E";M
 80 INPUT S
 90 IF S<0 OR S>M GOTO 70
100 PRINT:PRINT"IMMETTI L' INDIRIZZO FINALE TRA";S;"E";M
110 INPUT F
120 IF F<S OR F>M GOTO 100
130 IF F<S GOTO 100
140 FOR I=S TO F
150 PRINT"INDIRIZZO = ";I;"   VALORE IN MEMORIA = ";PEEK(I)
160 NEXT I
170 PRINT:PRINT"DESIDERI VEDERE ALTRA MEMORIA?"
180 INPUT A$
190 IF A$="SI" GOTO 70
200 END
```

PLOT (=disegnare una trama). Accende sullo schermo un punto luminoso di coordinate X,Y. Il colore del punto è predeterminato dall'istruzione COLOR seguita dal numero di codice colore. PLOT (e anche COLOR) non è comune a tutti i computer. È utilizzato per realizzare grafici, disegni e opere di computer art.

6Ø COLOR 4

7Ø PLOT 5,3

POKE (to poke=mettere, inserire). Permette di scrivere direttamente in una cella di memoria un valore (compreso tra O e 255) che può rappresentare, in codice ASCII, un dato o una istruzione. Per esempio POKE 15000,65 pone nella cella 15000 la lettera A (65 è il codice ASCII di A).

"SCRIVERE" NELLE MEMORIE

Conoscendo gli indirizzi (che dipendono dalla mappa delle memorie) e i valori giusti (sempre compresi tra O e 255) potrete far fare al computer le cose piu diverse. Con POKE, per esempio, potreste produrre musica o disegni sullo schermo. Andando a caso può succedere di tutto, anche che il computer si fermi. Niente paura. In caso di problemi spegnetelo e ricominciate da capo oppure passate ad un altro programma.

```
 10 REM SCRITTURA NELLA MEMORIA
 20 CLS:PRINT:PRINT
 30 PRINT"QUANTI K-BYTE DI MEMORIA HA IL TUO COMPUTER ?"
 40 INPUT KB
 50 B=KB*1024-1
 60 PRINT"IMMETTI LA LOCAZIONE IN CUI VOI SCRIVERE TRA O E";B
 70 INPUT L
 80 IF L<O OR L>B GOTO 60
 90 PRINT"IMMETTI IL VALORE CHE VUOI SIA MESSO ALL' INDIRIZZO";L
100 PRINT"COMPRESO TRA 0 E 255"
110 INPUT V
120 IF V<O OR V>255 GOTO 90
130 POKE L,V
140 END
```

PRINT (=stampa). Dice al computer di "scrivere" sullo schermo parole, numeri, risultati di operazioni e di equazioni. PRINT può visualizzare anche il contenuto di variabili numeriche (A,B, ecc.) e di variabili di caratteri (A$, B$, ecc.)

UN TRIANGOLO CONTROVERSO

C'è chi dice che gli arabi lo conoscevano già. C'è chi ne attribuisce la scoperta all'Italiano Tartaglia e chi al francese Pascal. È un triangolo formato da numeri con singolari proprietà. Con questo programmino potrete crearlo e studiarlo con il computer.

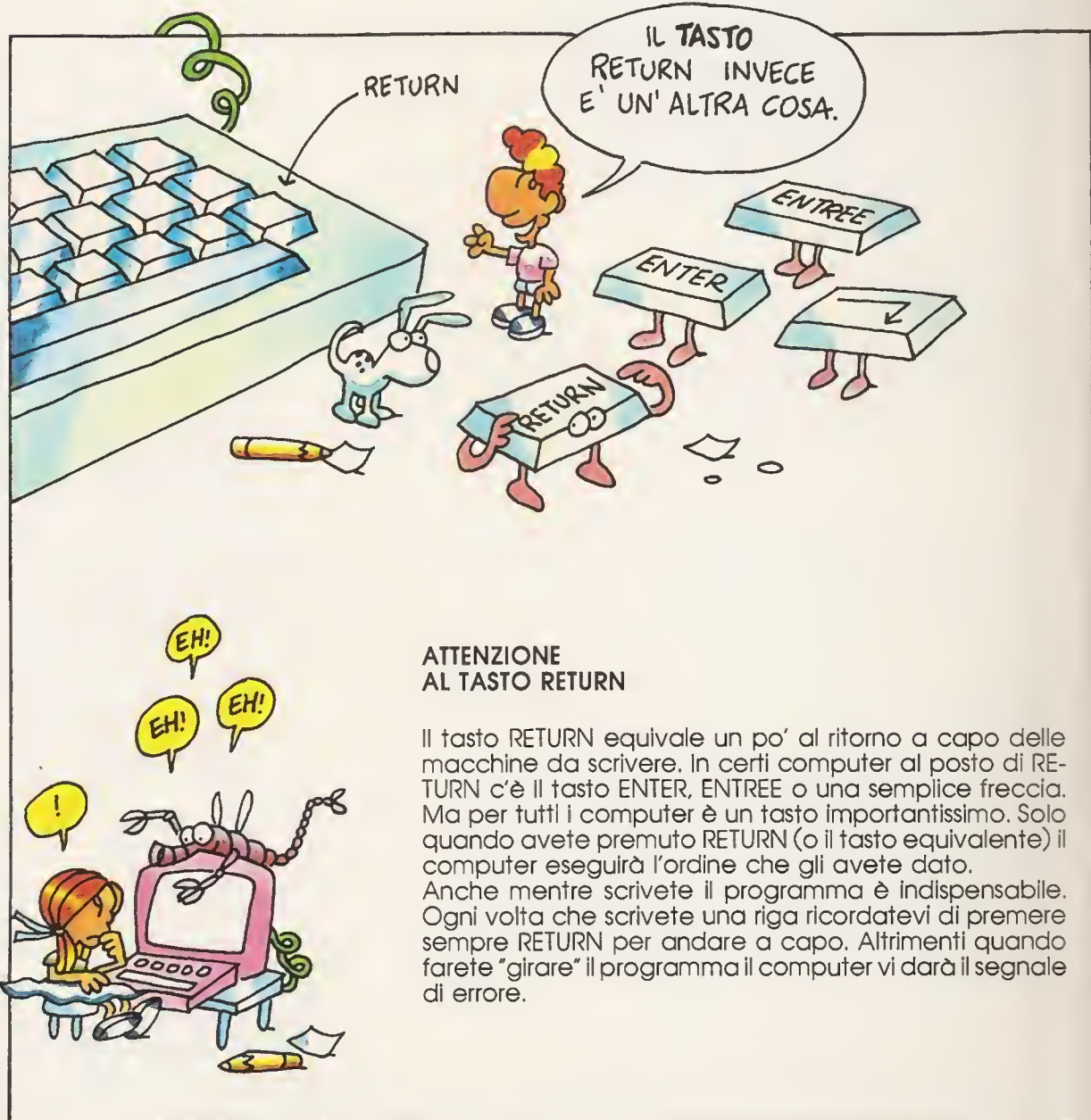
```
 10 REM IL TRIANGOLO DI TARTAGLIA
 20 CLS:PRINT"QUANTE RIGHE VUOI?"
 30 INPUT N
 40 DIM P(N,N)
 50 P(1,1)=1
 60 REM STAMPO IL PRIMO ELEMENTO
 70 PRINT:PRINT 1
 80 FOR K=2 TO N
 90 P(K,1)=1
100 REM LA PRIMA COLONNA = 1
110 PRINT P(K,1);
120 FOR J=2 TO K
130 REM PREPARO GLI ALTRI ELEMENTI
140 P(K,J)=P(K-1,J)+P(K-1,J-1)
150 PRINT P(K,J);
160 NEXT J
170 PRINT
180 NEXT K
190 END
```

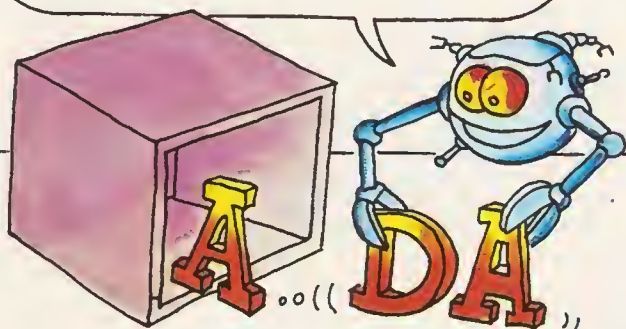
REM (da REMark=annotare). Dopo REM si possono inserire commenti utili per chi scrive o legge il programma. Quando il programma è partito se il computer incontra REM non fa nulla e passa alla riga successiva.

I NUMERI PERFETTI

Si dice "numero perfetto" un numero che è uguale alla somma di tutti i suoi divisori. Per esempio 6 è un numero perfetto, perchè i suoi divisori, 3, 2 e 1 danno per somma 6. Ecco un programma che cerca i numeri perfetti tra 1 ed N.

```
 10 REM I NUMERI PERFETTI
 20 REM 1 E' IL MINORE DEI NUMERI PERFETTI
 30 CLS:PRINT"IMMETTI IL NUMERO FINO A CUI CERCARE"
 40 INPUT N
 50 FOR K=1 TO N
 60 REM PONGO S=1
 70 S=1
 80 REM CALCOLO K/2
 90 D=INT(K/2)
100 FOR J=2 TO D
110 REM CERCO I DIVISORI DI K
120 IF K/J<>INT(K/J) GOTO 150
130 REM SE J E' UN DIVISORE LO SOMMO
140 S=S+J
150 NEXT J
160 REM SE K<>S ALLORA NON E' PERFETTO
170 IF K<>S GOTO 210
180 REM STAMPO IL NUMERO K
190 PRINT"NUMERO PERFETTO :";K
200 REM STAMPO UN MESSAGGIO
210 IF K/10=INT(K/10) THEN PRINT"STO CALCOLANDO"
220 REM CONSIDERO IL NUMERO SUCCESSIVO
230 NEXT K
240 REM IL PROGRAMMA E' FINITO
250 END
```

RETURN (=ritorno). È una istruzione BASIC che si usa soltanto per far rientrare il computer da un sottoprogramma al programma principale. In questo libro l'abbiamo usato nel gioco del NIM (vedi GO SUB).

ATTENZIONE AL TASTO RETURN

Il tasto RETURN equivale un po' al ritorno a capo delle macchine da scrivere. In certi computer al posto di RETURN c'è il tasto ENTER, ENTREE o una semplice freccia. Ma per tutti i computer è un tasto importantissimo. Solo quando avete premuto RETURN (o il tasto equivalente) il computer eseguirà l'ordine che gli avete dato.
Anche mentre scrivete il programma è indispensabile. Ogni volta che scrivete una riga ricordatevi di premere sempre RETURN per andare a capo. Altrimenti quando farete "girare" il programma il computer vi darà il segnale di errore.

RIGHT$ (right=destra). Dice al computer di estrarre la parte più a destra di una stringa di caratteri. Se P$="ADA", RIGHT$ (P$,2) è "DA". Nello Spectrum RIGHT$ va omesso e dovremmo scrivere P$ (2 TO 3). Nell'esempio sottostante RIGHT$ serve per estrarre la parte più a destra della parola che volete anagrammare.

GLI ANAGRAMMI

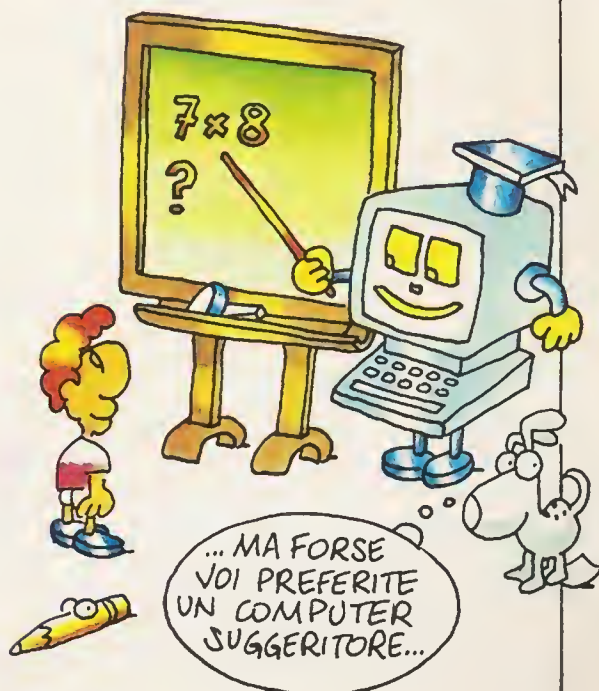
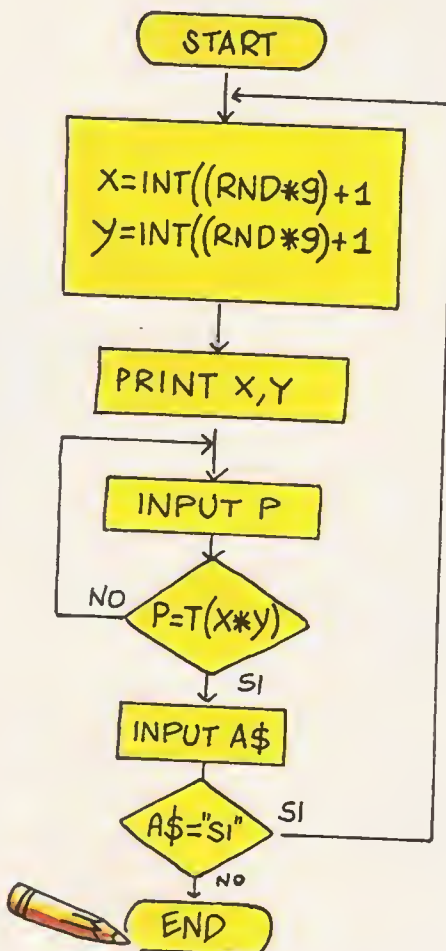
Anagrammare una parola vuol dire scambiare fra loro le sue lettere in tutti i modi possibili, per ottenere altre parole. Con questo programmino, limitato alle parole di tre lettere, il computer lo farà automaticamente.

```
 10 REM ANAGRAMMI DI UNA PAROLA DI TRE LETTERE
 20 DIM A$(3)
 30 CLS:PRINT"IMMETTI UNA PAROLA DI TRE LETTERE"
 40 INPUT P$
 50 REM METTO LE LETTERE NEGLI ELEMENTI DI A$
 60 PRINT
 70 FOR K=1 TO 3
 80 A$(K)=LEFT$(P$,1)
 90 P$=RIGHT$(P$,3-K)
100 NEXT K
110 REM ANAGRAMMA
120 FOR K=1 TO 3
130 FOR I=1 TO 3
140 IF I=K GOTO 170
150 J=6-(K+I)
160 PRINT A$(K);A$(I);A$(J)
170 NEXT I
180 NEXT K
190 END
```

RND (da random=casuale). Dice al computer di generare un numero "a caso". Il numero "estratto" è compreso tra 0 e 1. Perciò per ottenere numeri casuali interi si deve usare anche la funzione INT, e moltiplicare RND per valori interi.

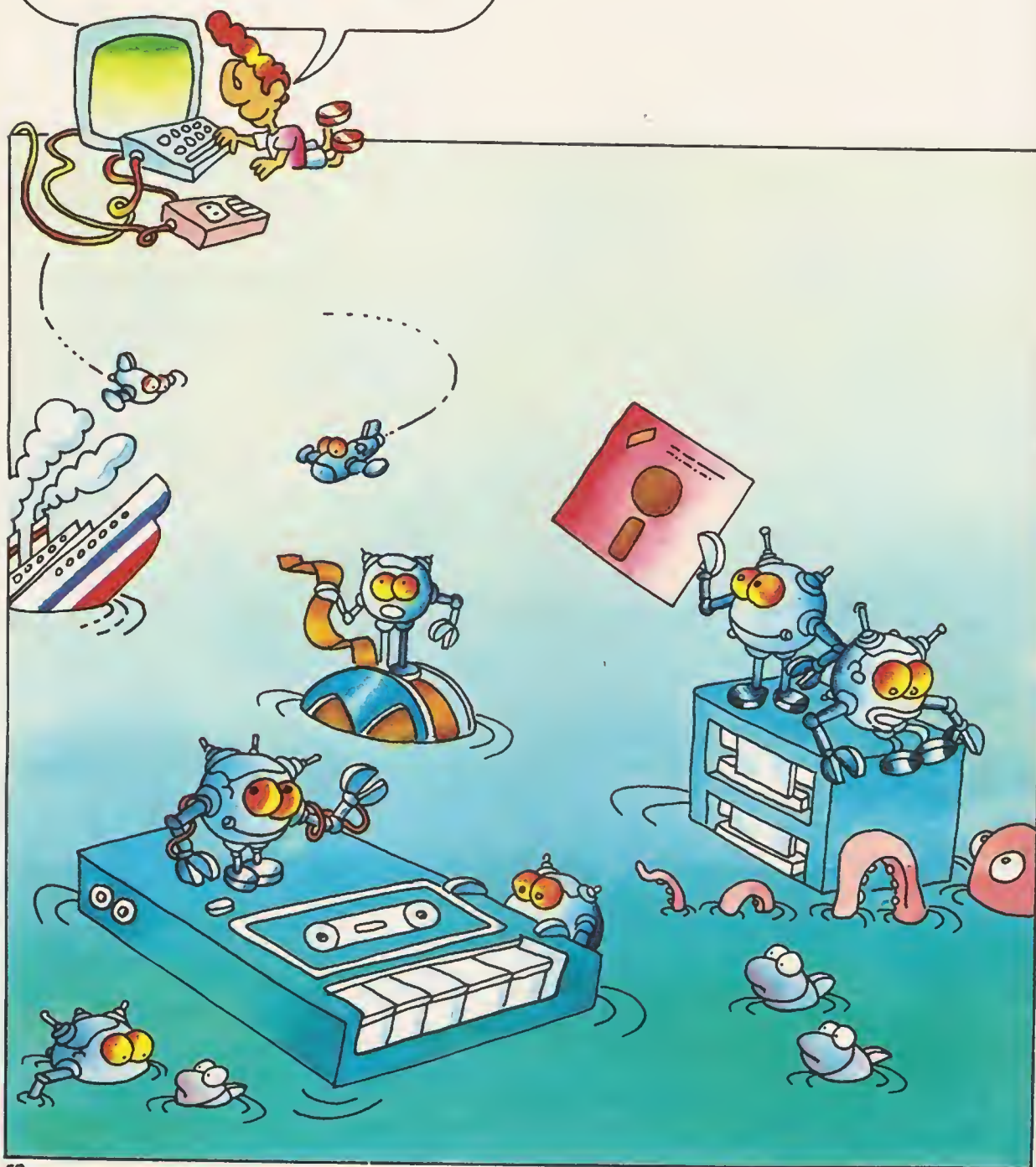
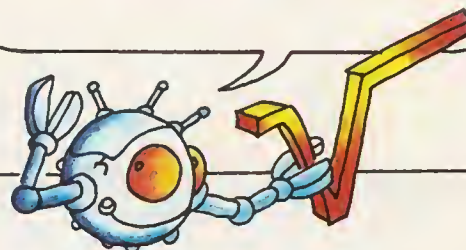
VOLETE UN COMPUTER PROFESSORE?

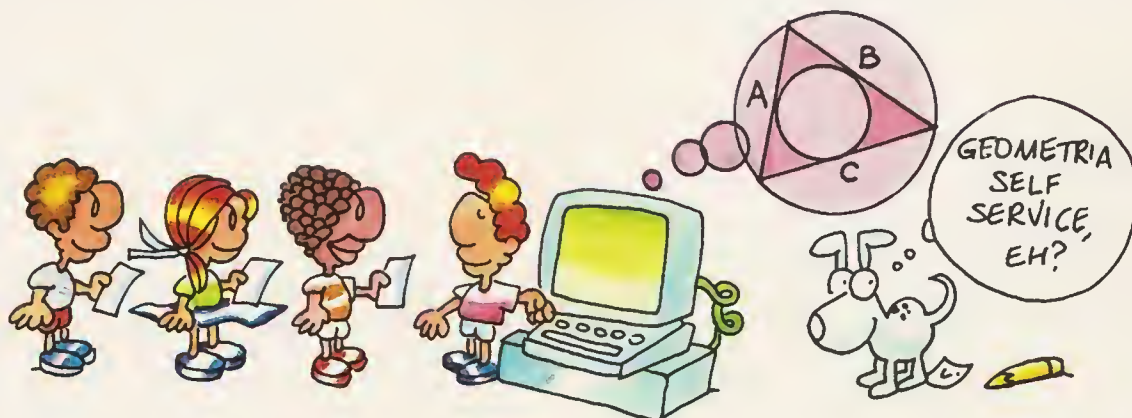
Con questo programmino il computer diventerà un professore di matematica che vi interrogherà sulle tabelline dall'1 al 9. Provate, anche se le tabelline le conoscete benissimo.

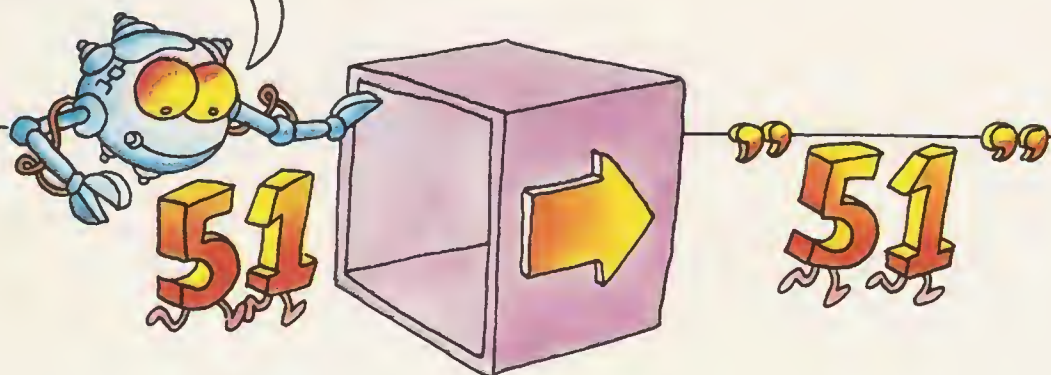
```
 10 PRINT"IMPARIAMO LE TABELLINE"
 20 CLS
 30 X=INT(RND*(9+1))
 40 Y=INT(RND*(9+1))
 50 PRINT"DIMMI IL PRODOTTO DEI DUE NUMERI  :"
 60 PRINT:PRINT X,Y
 70 PRINT:INPUT P
 80 IF P=X*Y GOTO 110
 90 PRINT:PRINT"RISPOSTA ERRATA. RIPROVA"
100 GOTO 70
110 PRINT:PRINT"RISPOSTA ESATTA.
                VUOI CONTINUARE? SI/NO"
120 INPUT A$
130 IF A$="SI" GOTO 20
140 END
```

RANDOMIZE (o RANDOM)

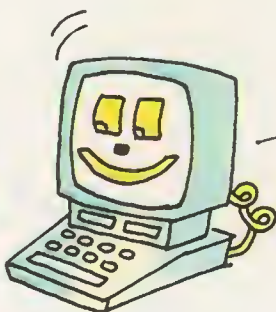
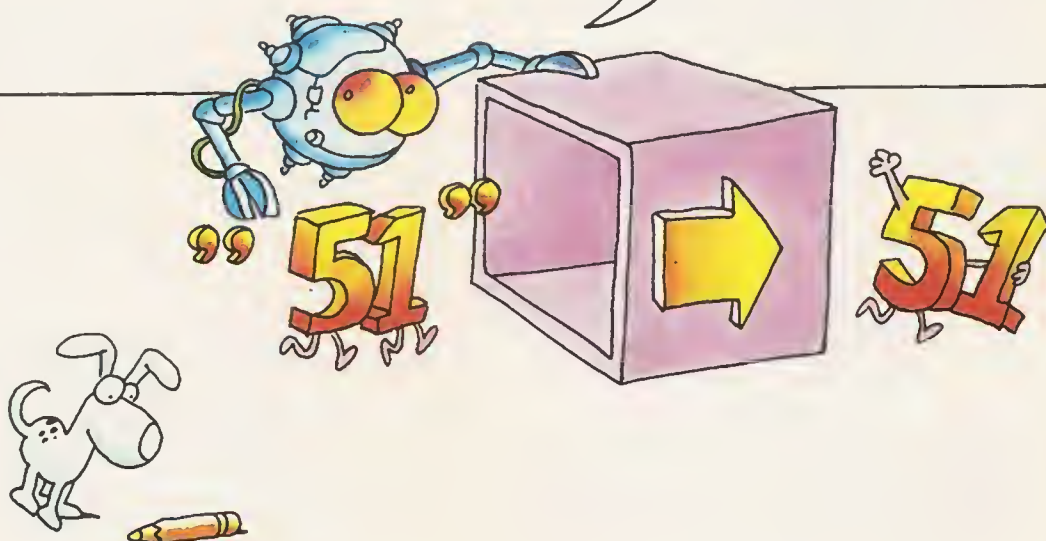
Pochi computer danno i numeri veramente casuali solo con RND. Infatti ad ogni giro di programma pescano dalle loro celle di memoria sempre le stesse sequenze di numeri. Per avere sequenze veramente diverse bisogna premettere l'istruzione RANDOMIZE o RANDOM, dipende dal "dialetto" che parla il vostro computer.

RUN (=gira). È l'ordine che fa "partire" il programma. Quando avete finito di scrivere un programma, o lo avete caricato in memoria, scrivete RUN e premete RETURN. Se tutto va bene il programma "girerà".

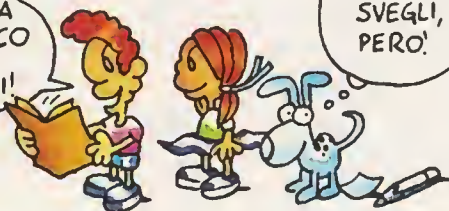
SAVE (to save=salvare, conservare). Ordina al computer di "mettere in salvo" il programma sulla cassetta del registratore o sui dischetti o sui nastri. Dopo SAVE si deve far seguire il nome del programma per poterlo riconoscere tra tanti altri, quando si vorrà richiamarlo.

SQR. Calcola la radice quadrata di numeri positivi. Per esempio SQR(4) è 2.

PROBLEMINO

Dati i lati di un triangolo qualsiasi ABC, trovare l'area del cerchio inscritto e del cerchio circoscritto. Ecco il programmino per risolverlo. Una volta per tutte.

```
 10 REM PROPRIETA' GEOMETRICHE DI UN TRIANGOLO
 20 CLS:PRINT"IMMETTI LE MISURE DEI LATI"
 30 INPUT A,B,C
 40 P=3.141592
 50 S=(A+B+C)/2
 60 A1=SQR(S*(S-A)*(S-B)*(S-C))
 70 PRINT:PRINT"AREA DEL TRIANGOLO = ";A1
 80 R1=A1/S
 90 A2=P*R1^2
100 PRINT:PRINT"AREA DEL CERCHIO MASSIMO INSCRITTO = ";A2
110 R2=A*B*C/(4*A1)
120 A3=P*R2^2
130 PRINT:PRINT"AREA DEL MINIMO CERCHIO CIRCOSCRITTO = ";A3
140 END
```

STR$ (da STRing=stringa). Dice al computer di trattare un valore numerico come stringa. STR$ (51) è "51".

LE EQUIVALENZE

Quante volte a scuola avete dovuto fare le equivalenze? Ora con questo programmino potrete farle fare al computer. Potrete immettere qualsiasi misura di peso e il computer usando STR$ e VAL estrarrà dalle stringhe i valori numerici essenziali ed effettuerà le equivalenze per vostro conto.

VAL (da value=valore). Dice al computer di trattare come valori numerici i numeri scritti come stringhe. Per esempio se A$="51", VAL (A$) è il numero 51. VAL è la funzione inversa di STR$.

```
 10 REM LE EQUIVALENZE
 20 CLS
 30 PRINT:PRINT"IMMETTI UNA MISURA DI PESO"
 40 INPUT A$
 50 N=VAL(A$)
 60 N$=STR$(N)
 70 L=LEN(N$)
 80 M=LEN(A$)
 90 A$=RIGHT$(A$,M-L)
100 A$=LEFT$(A$,3)
110 IF A$="GRA" THEN N=N*1:GOTO 170
120 IF A$="ETT" THEN N=N*100:GOTO 170
130 IF A$="KIL" THEN N=N*1000:GOTO 170
140 IF A$="QUI" THEN N=N*100000!:GOTO 170
150 IF A$="TON" THEN N=N*1000000!:GOTO 170
160 PRINT:PRINT"NON CONOSCO QUESTA UNITA' DI MISURA":GOTO 30
170 PRINT:PRINT"IN CHE UNITA' VUOI CHE SIA ESPRESSO IL RISULTATO"
180 INPUT B$
190 B$=LEFT$(B$,3)
200 IF B$="GRA" THEN N=N/1:C$="GRAMMI":GOTO 260
210 IF B$="ETT" THEN N=N/100:C$="ETTOGRAMMI":GOTO 260
220 IF B$="KIL" THEN N=N/1000:C$="KILOGRAMMI":GOTO 260
230 IF B$="QUI" THEN N=N/100000!:C$="QUINTALI":GOTO 260
240 IF B$="TON" THEN N=N/1000000!:C$="TONNELLATE":GOTO 260
250 PRINT:PRINT"NON CONOSCO QUESTA UNITA' DI MISURA":GOTO 170
260 PRINT:PRINT"IL RISULTATO ESPRESSO IN ";C$;" E':"; N
270 PRINT:PRINT:PRINT"DESIDERI FARE UN' ALTRA EQUIVALENZA ?"
280 INPUT R$
290 IF R$="SI" GOTO 20
```

COME SI USA QUESTO LIBRO

1. PER PRIMA COSA SI SFOGLIA E SI LEGGE. SCOPRIRETE CHE IL BASIC NON E' UNA "LINGUA" TANTO DIFFICILE.

2. POI PROCURATEVI UN COMPUTER, UN COMPUTER QUALUNQUE, IL VOSTRO, QUELLO DI PAPA', DELLA SCUOLA, DI UN AMICO...

3. RISOLTO IL PROBLEMA DEL COMPUTER COPIATE I "PROGRAMMINI" DEL LIBRO COMINCIANDO DA QUELLI CHE VI SONO PIÙ SIMPATICI.

4. MENTRE COPIATE UN ESERCIZIO RICORDATEVI SEMPRE, ALLA FINE DI OGNI RIGA, DI PREMERE **RETURN** (PAG. 53).

INVECE, PER FAR "PARTIRE" IL PROGRAMMA, DOVETE SCRIVERE **RUN** (PAG. 57) E POI PREMERE RETURN.

5.

SE APPARE UN MESSAGGIO DI ERRORE SCRIVETE **LIST** (PAG. 40), PER RICHIAMARE LA RIGA SBAGLIATA, O TUTTO IL PROGRAMMA.

6. CORREGGETE LE RIGHE SBAGLIATE, RISCRIVETE RUN, PREMETE RETURN E... ALLA FINE IL PROGRAMMA "GIRERÀ".

7. IN QUESTO LIBRO LE "PAROLE" DI BASIC SONO POSTE IN ORDINE ALFABETICO IN MODO DA FORMARE UN VERO E PROPRIO VOCABOLARIO.

COSÍ QUANDO INCONTRERETE UNA ISTRUZIONE O UN VOCABOLO CHE NON CONOSCETE POTRETE TROVARLO FACILMENTE NEL "VOCABOLARIO" OPPURE NELL' INDICE ANALITICO (PAG.64).

8. SE QUALCHE PROGRAMMA INSISTE AD ESSERE UN PROBLEMA AIUTATEVI COL MANUALE DEL VOSTRO COMPUTER. POSSONO ESSERCI PICCOLE DIFFERENZE TRA IL "DIALETTO BASIC" DEL VOSTRO COMPUTER E QUELLO DEI NOSTRI PROGRAMMI. UN CONFRONTO SARA' UTILE.

9.

ALCUNI PROGRAMMI SONO SEGUITI DAI LORO "DIAGRAMMI DI FLUSSO". QUESTI SCHEMI RIPORTANO SOLO I PASSAGGI LOGICI FONDAMENTALI, I"PRINT" E I"REM" NON SONO RAPPRESENTATI. CONFRONTATELI CON I PROGRAMMI, IMPARERETE L' ARTE DI PROGRAMMARE.

10. GLI "ESERCIZI" SONO PER TUTTI I GUSTI. ALCUNI SONO "SCOLASTICI" ALTRI SONO GIOCHI DIVERTENTI. POTRETE "SALVARE" QUELLI CHE VI PIACCIONO DI PIÙ CON **SAVE** (PAG.58) E COMINCIARE COSÌ LA VOSTRA BIBLIOTECA DI PROGRAMMI.

INSOMMA SE USERETE BENE QUESTO LIBRO ALLA FINE SAPRETE ABBASTANZA BASIC DA FARVI INVIDIARE DA PAPÀ.

INDICE ANALITICO

Questo volume è stato impresso
nel mese di giugno dell'anno 1984
presso le Arti Grafiche delle Venezie di Vicenza
Gruppo Mondadori

Stampato in Italia - Printed in Italy

In questo volume Ada e Marko in compagnia
del cagnolino Plotter e di un personal computer
allegro e cortese riprendono il viaggio
colorato ed esilarante attraverso l'informatica
cominciato con **Il mio primo libro sui computer**
dello stesso autore, Luca Novelli.

Il BASIC è un linguaggio inventato dagli informatici
per dire al computer quello che deve fare,
in altre parole, per scrivere dei 'programmi'.
L'autore lo ha trasformato
in un grande gioco, affascinante e divertente.
Un tipico programma BASIC diventa così un variopinto
'gioco dell'oca' dove si possono percorrere
itinerari tortuosi o fulminei, scorciatoie astute,
andirivieni estenuanti e passaggi al volo disseminati
di insidiose trappole (i famosi 'bug' o errori di programmazione).

Il libro è un vero e proprio vocabolario
di BASIC e per ogni vocabolo c'è un programma
o un gioco. È un invito per i giovani
e per i più piccoli a essere creativi
con il computer, a casa e a scuola.

Ma è anche un invito rivolto a tutti coloro che desiderano
essere introdotti nel mondo
della programmazione e dei 'linguaggi' per parlare con il computer.
È un libro che si può leggere tutto d'un fiato,
ma soprattutto è un amico da tenere
vicino al computer per 'imparare facendo'.

Luca Novelli, popolare autore di strisce a fumetti, dal 1977 scrive
e disegna libri di divulgazione scientifica per ragazzi.

Nella stessa collana: